
INPUT
Publicación práctica para usuarios de
sinclair
Octubre 1985 Precio 350 Ptas
Año 1 Número 2
Contiene un coleccionable
NO LO OLVIDES ¡ARCHIVALO!
PROTEGE TUS PROGRAMAS
LA CAZA DEL CORRECAMINOS
C-20
CINTA PARA ORDENADOS
PC15
TDK
POSTER
PANORAMA PARA MATAR



¿Y AHORA...?
¡Tan tranquilo! Gracias al "Coste estándar por reparación" de
HISSA
Servicio Oficial
investronica
no debe preocuparte ni el niño de la casa, ni siquiera un terremoto.
Con HISSA tienes la certeza que tu ordenador Spectrum es atendido por expertos técnicos y reparado con piezas originales SINCLAIR. ¡Y sin sobresaltos con el precio!
"COSTE ESTANDAR POR REPARACION"
ZX 81: 3.150 Ptas.
Spectrum 16K: 5.250 Ptas.
Spectrum 48K: 6.300 Ptas.

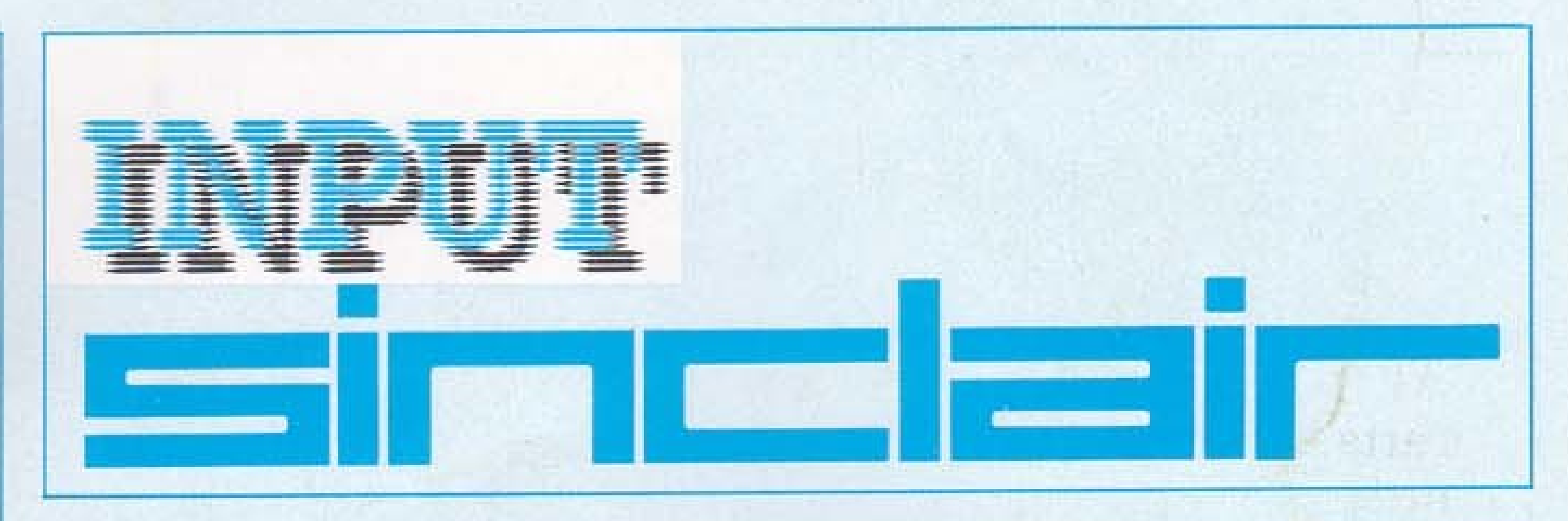


DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Antonio Taratiel, Luis R. Palencia, Christophe Pais, Francisco Tórtola, José Pérez Montero, Benito Román
INPUT Sinclair es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: Grupo Jota
Madrid: c/ Cuenca, 1, 1.°
Tel. 253 45 01/02
Barcelona: Avda. de Sarriá, 11-13, 1.°
Tel. 250 23 99
FOTOMECANICA: Ochoa, S.A.
COMPOSICION: Fernández Ciudad, S.L.
IMPRESION: Edime, S.A.
Depósito legal: M. 27.885-1985
Suscripciones: EDISA,
López de Hoyos, 141. 28002 Madrid
Tel. (91) 415 97 12
Redacción:
Alberto Alcocer, 46, 4.°
28016 Madrid. Tel. 250 10 00
DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona
Se solicitará el control OJD
INPUT Sinclair es independiente y no está vinculada a Sinclair Research o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.



Sumario

AÑO 1 NUMERO 2

VUESTRAS CARTAS

Parece que hemos entrado con buen pie. Al menos así lo indica el gran volumen de cartas recibidas. Ocupa el primer lugar, por orden de importancia, los envíos a **Los mejores de INPUT.** Tal es así que aumentaremos la cantidad de cintas a sortear. En el próximo número publicamos la lista de premiados, aunque las cintas ya están en camino.

Vuestros artículos y programas, para la sección **Participa,** comenzaron a llegar tímidamente los primeros días, pero acaba de producirse una verdadera avalancha.

Enviáis cartas de contenido realmente halagador en un noventa y nueve por ciento, pero hay un uno por ciento que incorporan una actitud de «¿Qué quieren vendernos estos señores?», vamos a ver en qué les cazamos». Pues bien, es lógico que haya pequeños fallos, achacables a la puesta en marcha de la revista, pero si se miran con lupa se ven más grandes. No olvidéis que escribimos para y no contra vosotros. En realidad, el equipo de redacción está formado por gente tan entusiasta de los micros como podáis serlo cualquiera.

El precio ha sido el tema de alguna carta. En realidad no está tan desfasado con respecto a otras revistas nacionales de ésta y otras especialidades. Por otro lado, las revistas extranjeras, que abundan en los kioscos, no son precisamente baratas. La calidad de impresión tiene sus costos y hemos ajustado los márgenes al máximo. Siempre ponemos como ejemplo a las publicaciones extranjeras y nuestra primera intención ha sido proporcionaros (a nosotros también) una revista de nivel comparable con ellas. Queremos seguir en ese camino. Vosotros, obligándonos a imprimir una Segunda Edición, parece que estáis igualmente de acuerdo.

EL BUZON DE INPUT

El próximo número de INPUT reservará un espacio para responder a vuestras cartas.

No pretendemos que sea un rincón únicamente destinado a la resolución de dudas y problemas. Por supuesto que intentaremos proporcionaros este tipo de ayuda, pero nuestra pretensión es tener algo más: un espacio para la comunicación donde nos contéis a todos lo que hacéis con el ordenador, proyectos, intenciones, cómo es vuestra instalación y las cien mil cosas que se os ocurran.

Dirigir las cartas a:

EL BUZON DE INPUT SINCLAIR
Alberto Alcocer, 46, 4.º B
28016 Madrid

MEJORAS Y APLICACIONES

Cualquier programa publicado en INPUT puede sugerir nuevas ideas. Te lanzamos una propuesta a modo de reto: mejora y/o encuentra aplicaciones a nuestros listados. No olvides que muchas veces salta una chispa que trae una solución genial.

Cada tercer mes, a partir de la publicación de un número, elegiremos la mejor aportación.

La decisión será inapelable, pero tener por seguro que será ecuánime y, sobre todo, resuelta por un grupo de expertos.

El ganador recibirá un premio consistente en una suscripción por un año a INPUT SINCLAIR.

¡PARTICIPA EN INPUT!

Si quieres ver tus programas, ideas, o artículos, publicados en tu revista, examina las bases y haznos llegar el material.

Publicar tiene su recompensa.

BASES

PROGRAMAS: Una vez desarrollado tu programa, que debe ser original y no haber sido enviado a ninguna otra publicación, puedes enviárnoslo aquí grabado en cassette, diskette o microdrive. Es preferible que vaya acompañado por un listado de impresora, pero no es imprescindible.

El programa habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

ARTICULOS E IDEAS: Se aplica lo anteriormente dicho para los textos que acompañan a los programas; es decir, conviene detallar al máximo lo que desees que aparezca publicado en la revista, de la manera que te gustaría que otra persona hubiera explicado eso mismo.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvidéis indicar claramente para qué ordenador está preparado el material, así como vuestro nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante cada mes SORTEAREMOS:

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
  en material microinformático a elegir por los afortunados.

¡No os desaniméis!, por muy simples o complejas que puedan parecer vuestras ideas, todas serán revisadas con el máximo interés.

INPUT SINCLAIR
Alberto Alcocer, 46, 4.° B
28016 Madrid

NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.



EL QL ATACA DE NUEVO

Sir Clive Sinclair nunca se rinde. Tras los recientes problemas habidos con su "salvador", Maxwell, acaba de anuciar increíbles bajadas de precio en sus productos.
Por ejemplo, está dispuesto a vender el QL a doscientas libras esterlinas, la mitad del anterior precio. En algunos comercios minoristas británicos se venderá un lote compuesto por un Spectrum Plus, diez cintas de programas, un dispositivo de *cassette*, el Interface 2 y un *joystick*, por ciento cuarenta libras.
La operación con la cadena de distribución Dixons absorverá un *stock* sobrante de ciento sesenta mil máquinas.
Por otra parte, el equipo científico que tiene sir Clive trabajando en el desarrollo del superchip en una oblea afirma estár en el buen camino, tras resolver los principales problemas tecnológicos previstos.



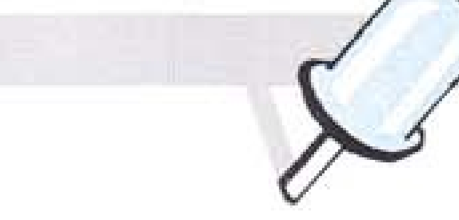
CASTELLANIZAR

Durante los últimos días se rumoreó insistentemente la posibilidad de que castellanizar los teclados y el *software* pasaría a ser una medida de cumplimiento obligatorio.
No quedaba claro si esto sería aplicable a todos los sistemas, solo a los adquiridos por las escuelas o deberían atenerse a ello quienes quieran vender a la Administración. De momento sigue siendo un rumor.



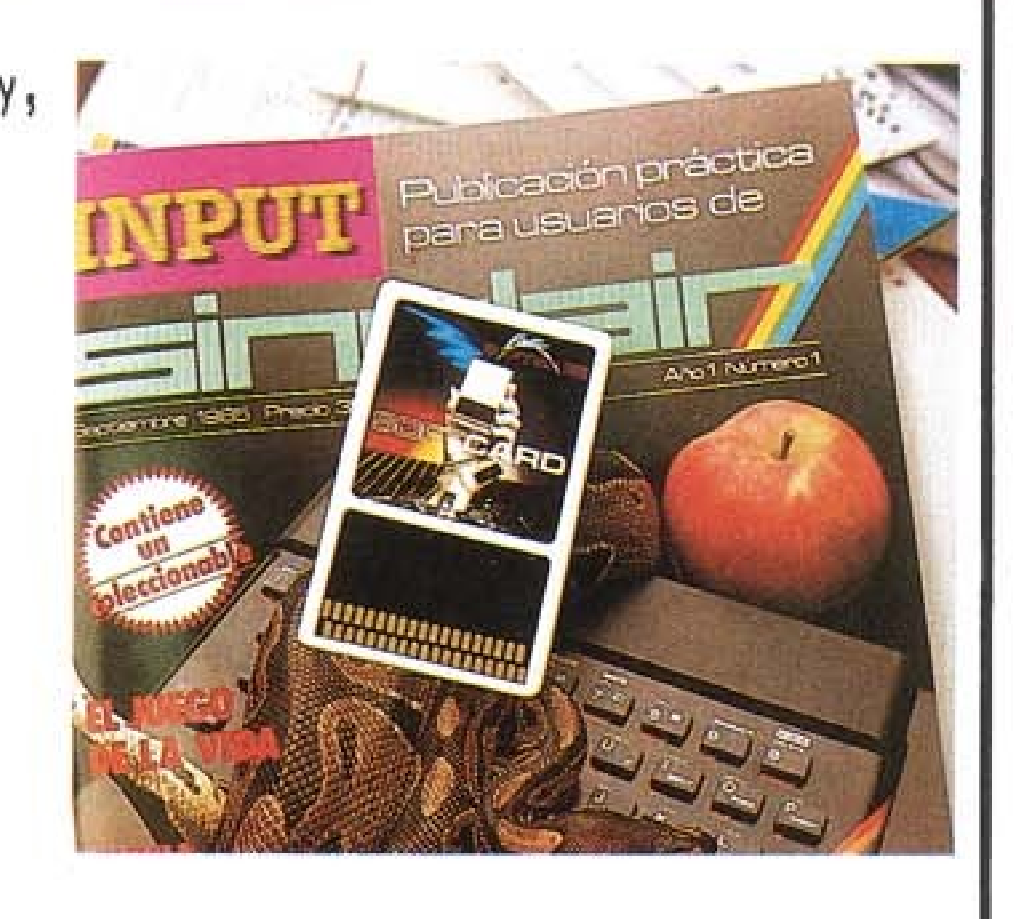
NUEVOS ARANCELES

Un caluroso día de julio algo sobresaltó a los importadores de microordenadores que abrieron el Boletín Oficial del Estado.
Una indiscriminada subida arancelaria lineal, de quince mil pesetas, afectaba a todos los sistemas cuyo precio estuviera por debajo de las trescientas mil.
La cosa creaba dudas, hasta el punto de que cualquier periférico o accesorio pudiera sufrir igual trato.
Esta situación se corrige a principios de septiembre, logicamente a través del B.O.E.
Indignación e impotencia es el sentimiento que inunda a una mayoría de los atónitos distribuidores.
Los intentos para hallar justificación a esta medida apuntan a la clásica protección de una industria nacional. Es escualida, por no decir nula, argumentan algunos y la especulación hace pensar en la llegada de fabricantes extranjeros, entre los que se barajan Fujitsu, de la mano de Secoinsa, y su modelo F-7. Tal vez sea Sinclair o Atari, dicen otros. Incluso hay quienes apuntan directamente a Dragon y su nuevo proyecto en Galicia como detonante.



LA TARJETA CREDISOFT

Parece una tarjeta de crédito y, sin embargo, es portadora de sueños. En su interior alberga un diminuto *chip* de ROM (memoria que solo puede leerse) en el que se ha dispuesto información, que muy bien puede ser el código máquina de un juego.
Al igual que ocurre con los *chips* de ROM clásicos, existen alternativas para las nuevas tarjetas. Una variante es la PROM, que es una memoria en blanco que puede ser grabada por el usuario, gracias a un dispositivo electrónico especial.
Otra es la EPROM, en la que pueden ser grabados los datos y posteriormente borrados para almacenar nueva información.
El desarrollo es fruto de la colaboración de dos empresas: Mitsubishi y Hudson Soft.
Las capacidades de la versión grabada por el fabricante cubren un rango situado entre los 64 Kits y 1 Mbit, mientras que las otras dos tienen, por ahora, como límite máximo los 256 Kbits.
Los ordenadores MSX ya disponen de un adaptador, con forma de cartucho, que permite la conexión directa
al bus del sistema. En breve estarán disponibles las versiones para Spectrum y los Commodore.
Las treinta y dos diminutas pistas eléctricas (recubiertas de



oro) se conectan a los respectivos *ports* del usuario en el ordenador, por medio del aludido adaptador.
Esta nueva modalidad de presentación del *software* pone mas alta la barrera a los "piratas", que deberán ser mas sofisticados.
El formato de la tarjeta es de ochenta y seis por cincuenta y cuatro milimetros, siendo uno con ocho su espesor.
Además, se puede imprimir un diseño gráfico o texto por ambas caras.
La firma distribuidora Serma ha firmado con Hudson para la comercialización de la Beecard en nuestro país.



UNA RAZON DE PRECIO

La subida lineal en las tasas arancelarias, que cogió por sorpresa a casi todos los importadores cuando fue publicado el correspondiente decreto en el BOE, es probable que no afecte- al menos de momento- a los ordenadores ZX Spectrum.
La razón obedecería a los insistentes rumores que tienden a apuntar que el distribuidor habia adquirido una importante partida de ordenadores poco tiempo antes de la "sorpresa". Incluso llega a barajarse la cifra de 50.000 unidades.
El volumen de la operación sería debido a los apuros económicos del fabricante británico, quien a principios del verano convocó a sus distribuidores, pidiendoles que anticipasen sus pedidos previstos para la campana navideña.
De confirmarse este rumor, habrian razones para que el precio en el mercado nacional se mantuviera estable durante algún tiempo.



SPECTRUM DE 128 K

Lo anticipabamos en nuestro anterior numero. Sinclair estaba próximo a anunciar un nuevo modelo, un sucesor para éxito del Spectrum. Es probable que cuando estas líneas lleguen a manos del lector el nuevo Spectrum de 128 K haya sido anunciado oficialmente en el Sonimag; como también lo es que únicamente se venda en nuestro país. La supremacia en número de unidades vendidas del Spectrum convierte a nuestro mercado en el más suculento de Europa en los planes de este fabricante.
Conviene recordar que en Inglaterra la compentencia entre diversas marcas es brutal y es, por tanto, más difícil destacar. En Francia existen muchos fabricantes nacionales que se distribuyen la participación de un modo más homogéneo.
Hasta ahora, los fabricantes han sentido una especial debilidad por desarrollar nuevos ordenadores que no tuvieran nada que ver con sus modelos anteriores, ni siquiera a nivel de compatibilidad del *software*. Esa medida, que en principio fue un argumento para conseguir mayores ventas, se ha convertido en un *boomerang* vuelto contra las marcas. Ahora se tiende a proporcionar más potencia, memoria y mejor precio, sin olvidar la compatibilidad. Claros ejemplos son el Commodore 128, compatible con el 64 y el nuevo 128 de Sinclair, que dispondrá (como es lógico pensar) de 128 Kbytes de memoria, pudiendo utilizar el *software* ya disponible cuando trabaja en modo Spectrum. Todo parece indicar que el nuevo modelo incorpora un segundo microprocesador para gestión de su mayor memoria y, además, dispone de un teclado de mejor calidad, que incluye teclas numéricas separadas.
Lo que aún está por determinar es qué casas de *software* desarrollaran los programas que den soporte al 128. Tal vez haya que recurrir a las pocas empresas españolas capaces de hacerlo.
Por otra parte, aún no se ha desvelado el misterio que envolvió a la visita de Jack Tramiel, presidente de Atari, al distribuidor de Sinclair en nuestro país. Se especula con la posible creación de una competencia calculada entre ambas marcas en un futuro (520 ST versus Sinclair 128), dificultando así la presencia de una tercera marca con fuerza en la liza. Esto, unido a los rumores de fabricación (de varios fabricantes nacionales y extranjeros) y el decreto que añade nuevos aranceles a la importación de microinformática de bajo precio, puede hacer que tengamos uno de los mercados interiores más competitivos



SOFTWARE EDUCATIVO

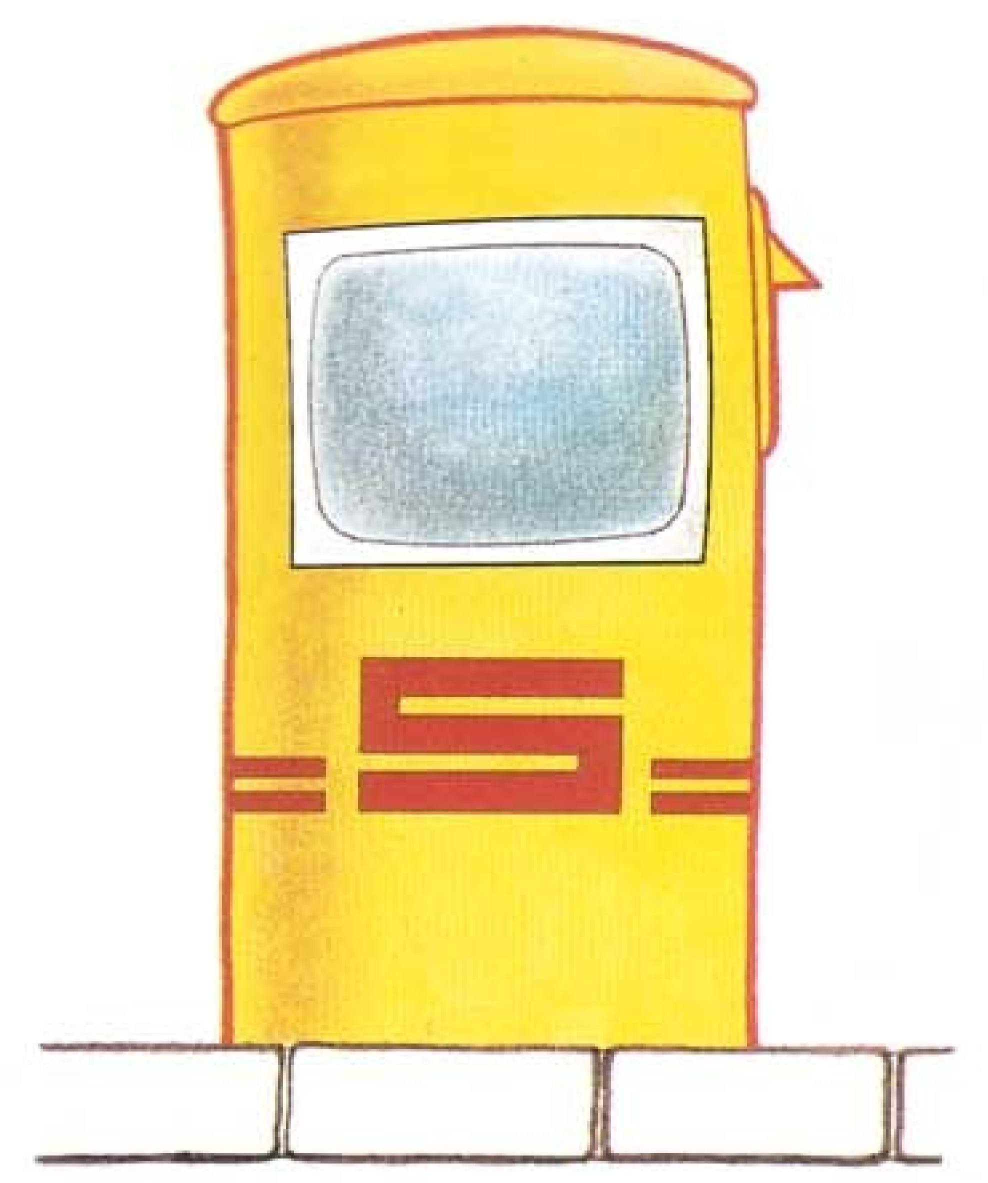
La ley de Boyle Mariotte es el primer tema elegido por Ediciones S.M.para una serie de programas educativos.
Esta editorial, especializada en textos destinados a escolares, acaba de lanzar el primer *cassette* destinado al Spectrum de 48 K. Viene complementado con un manual explicativo.
Los próximos titulos previstos son: Refracción de la luz, Planos inclinados y Corriente continua.



COMIENZA LA GUERRA

A mediados de julio, la lucha contra la copia ilegal de *software* recibió un espaldarazo definitivo en Gran Bretaña.
En efecto, una antigua ley de 1956 sobre el *copyright* ha sido rectificada, observandose en ella los derechos sobre el *software* para ordenadores.
En adelante serán severamente castigados quienes sean hallados culpables de duplicar, distribuir o impotar *soft* sin los debidos permisos.
La multa será determinada por el tribunal, sin limitaciones y el culpable podría ir a la carcel hasta dos años.
Los que tengan en su poder o vendan programas "piratas" podrán verse multados con hasta dos mil libras esterlinas por cada delito.

EL BUZON DE INPUT



Quisiera saber qué es el ATTR del código máquina.

Carlos Granados. Madrid

Hasta donde nosotros sabemos, en el léxico del Spectrum, ATTR significa ATTRibutos y no es código máquina, aunque sí puede emplearse para utilizarlo en este tipo de programación desde el BASIC.

Como sabes, la pantalla se divide en cuadraditos imaginarios sobre los que son visualizados los carcteres. Cada uno de esos cuadrados lleva asociado unos atributos que son: los colores del papel, la tinta, el brillo y la intermitencia. Con ellos se personaliza el aspecto de cada carácter visualizado.

Lo que más te puede interesar es que ATTR proporciona la información sobre un determinado cuadrado. Por ejemplo, PRINT ATTR (x,y), donde x e y son las coordenadas del carácter en cuestión, te proporcionará un número comprendido entre 0 y 255, que, traducido a binario, es un número de ocho cifras, que serán cada una de ellas un cero o un uno. Los tres primeros bits se refieren al color de la tinta (ocho colores en total), los tres siguientes bits son el color del papel, el séptimo bit dice si el carácter es visible o está «apagado» y el octavo informa si el carácter parpadea cuando su valor es 1.

Refiriéndome al artículo publicado con el nombre de «Mensajes en clave-secreto», quisiera formular una pregunta y es la siguiente: Si existe un programa elaborado para el Spectrum **de «Código de Acceso», en el cual sin la palabra clave no se puede tener acceso (valga la redundancia) a un programa o a un menú de distintos programas.**

Encarnación Paredes. Sevilla

En realidad, el tipo de protección que se requiere para evitar el acceso a datos o programas no tiene directamente que ver con la cifra de mensajes secretos, aunque un programa de esta clase pueda también incorporarse como rutina en uno más completo de protección.

En segundo lugar hay que definir el tipo de protección que deseamos. Puede ser del programa completo contra la copia. En tal caso, el artículo «Protege tus programas» ofrece una serie de valiosas pistas..

Si únicamente deseamos que no tengan acceso personas no familiarizadas con la informática, basta con introducir estas líneas de programa en el lugar adecuado:

```
999 a$ = «Palabra clave»
1000 INPUT a$: IF a$<> «Palabra
  clave» THEN GOTO 1000
```

La palabra clave la define el propio programador y puede cambiarla cuantas veces desee. Llegado a este punto, el programa se detendrá en la línea 1000 (los números de línea pueden ser cualesquiera) hasta que no se escriba correctamente la clave de acceso.

Supongamos que cargo de un *cassette* un programa escrito en código máquina, pero sin cabecera. ¿Cómo puedo averiguar en qué posición de memoria comienza y cuál es la longitud de dicho programa? Y otra cosa, para quienes les gustan los trucos y para aquellos que lo han probado casi todo, que tecleen POKE 23659,1, agárrense y pulsen ENTER.

Y, aparte, ¿qué podría hacer para que las teclas del ordenador no se borren con el uso?

Fco. Javier Almenar. Valencia

Muchas gracias por las felicitaciones. Es norma de ***INPUT*** *no publicar ese párrafo que soléis escribir todos, por dos cosas, si son elogios nos da vergüenza y si no lo son..., bueno hasta ahora no se ha dado el caso. ¿Será de preocupar?*

Bromas aparte, tu primera pregunta deja muy clara las intenciones que esconde, pero para que no digas que somos malos, una pista. En la página 100 de un libro comentado en anteriores ediciones de la revista se incluyen datos más que suficientes para un iniciado.

Lo de tu POKE, ¡sensacional oye!

Las leyendas de las teclas van serigrafiadas sobre la goma. Aunque la tinta resiste bastante, es cierto que puede llegar a borrarse. Una solución sencilla y poco atractiva consiste en meter el ordenador dentro de una bolsa de plástico transparente, que te permite ver las teclas, pero no alterar demasiado la pintura. De todas formas, no estaría de más escribir al distribuidor, por si dispone de kits de recambio o etiquetas adhesivas.

Desearía preguntarles si hay algún método para que no ocurra lo siguiente: Tengo el juego «Fútbol Manager» para el Spectrun. **Al ponerlo y empezar a jugar, al cabo de un rato me pone un mensaje en la pantalla que dice «c NONSENSE IN BASIC, 8032:1», y así siempre que pongo dicho juego. Por lo que nunca puedo jugar. Si saben la solución, por favor, díganmela.**

Alberto J. Martínez. Madrid

Las dos causas que nos parecen más probables son que la cinta tenga un defecto de grabación, con lo que el comercio que lo vendió no pondrá pegas en cambiártelo por otra copia, o un desajuste de la cabeza de tu cassette con respecto a la pista donde va grabado el programa. Puedes probar con un destornillador, ajustando el «azimut» (inclinación) de la cabeza del reproductor, dándole media vuelta en uno y otro sentido al tornillo que la sujeta.

IMPRESORAS: LA ELECCION

- ¿NECESITAS UNA IMPRESORA?
- ELECCION DEL TIPO CORRECTO
- COMPRANDO EL PAPEL
- CONEXION DE LA IMPRESORA AL ORDENADOR

Si alguna vez has deseado tener una copia de tu programa o un volcado de pantalla de algún gráfico, es señal de que necesitas una impresora. Pero, ¿cómo averiguar qué tipo comprar?

La compra de una impresora es una forma de extender la utilidad de tu ordenador y de desarrollar tu habilidad de programación. Pero puede resultar una inversión muy costosa, y como las posibilidades y el margen de precios son tan variados como los ordenadores, solamente el micro-usuario mejor informado estará seguro de la necesidad de una impresora o de comprar la más adecuada.

La forma normal en que un ordenador presenta información al usuario es sobre una pantalla de televisión o sobre un monitor de vídeo. La impresora es una forma alternativa de presentar la misma información. Pero no pienses que ambos dispositivos tienen los mismos usos, ya que uno no es sustituto del otro. La principal misión de una impresora es entregar una copia o registro permanente *(hard copy)* de cualquier información que pueda aparecer en la pantalla.

Por ejemplo, hay muchos atractivos diseños gráficos —compu-arte, diríamos— que puedes imprimir, archivar e incluso construir, pero sin una impresora sólo existen de una forma efímera sobre la pantalla de un monitor. Si, por ejemplo, escribes un programa largo, es posible que quieras una copia del listado para trabajar sobre él fuera del ordenador. Si escribes o

Una sola columna de agujas sin punta golpea una cinta entintada para crear el conjunto de puntos que forman cada carácter, en una impresora de matriz de puntos.

compras programas comerciales, necesitarás imprimir gráficos o cartas para ser incluidos en los informes. Y para el diseño asistido por ordenador necesitas un registro de formas u objetos al que poder hacer referencia desde el ordenador.

Todos los ordenadores domésticos tienen un teclado, que es semejante al de una máquina de escribir, de modo que puedes escribir y editar correspondencia, datos e incluso un libro, como sobre una máquina de escribir. Tu capacidad de manejar textos puede ir mucho más allá de lo que es meramente escribir a máquina si te sumerges en los dominios del proceso de textos, que te permiten manipular textos de múltiples formas. Con el *software* adecuado, tu micro tiene ya esta posibilidad, pero un bloque de texto muy cuidado, ya sea una carta o el contenido de un informe, es de poca utilidad si no se puede imprimir.

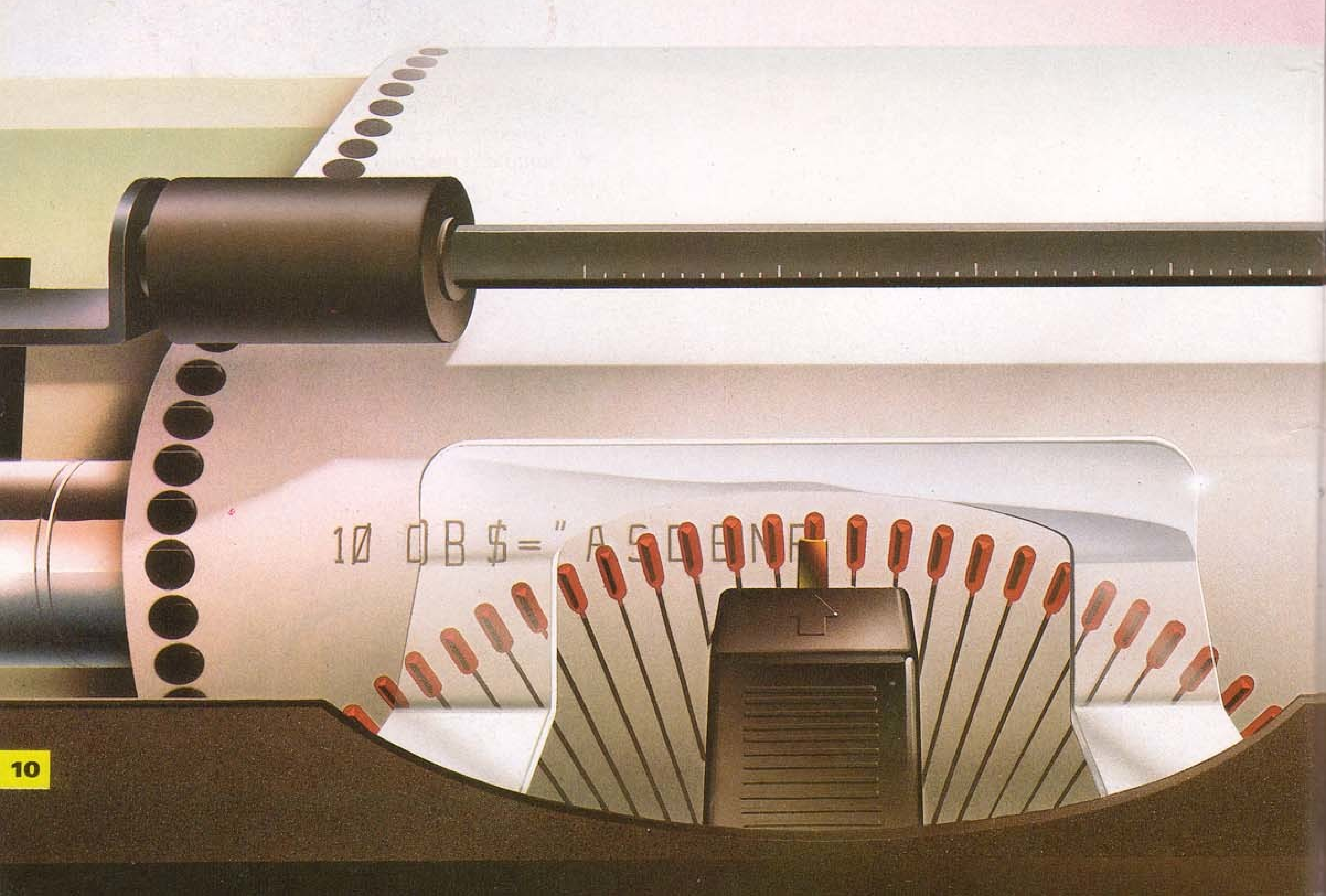
Y casi todos los usuarios encontrarán que una impresora es de gran utilidad como ayuda de programación. Además de proporcionar una copia de seguridad *(back-up)* de tu trabajo, con frecuencia es más fácil detectar errores sobre un papel que haciendo correr el texto del programa sobre la pantalla de un monitor. Un monitor tiene muchas ventajas, pero, en general, la gente prefiere la información impresa a la imagen electrónica, ya que le resulta más familiar.

Todas estas características pueden hacer que una impresora aparezca como una ayuda esencial para los usuarios del ordenador. Pero hay que decir que no todas las impresoras tienen todas estas posibilidades, y que las impresoras más versátiles tienen un precio prohibitivamente caro. Por ejemplo, no todas las impresoras pueden manejar gráficos, y sólo unas cuantas pueden imprimir en más de un color, normalmente (aunque no siempre) a un alto precio. Por esta razón es de la mayor importancia evaluar bien tus necesidades y elegir adecuadamente. El tipo que elijas depende de la aplicación que quieras darle. Para facilitar tu elección, empieza por la consideración de las características de los distintos sistemas disponibles.

TIPOS DE IMPRESORAS

Hay cuatro tipos de impresoras que podrían ser útiles para el usuario doméstico de un ordenador. Son la impresora por impacto de matriz de puntos, la de margarita, la térmica y la de plumilla.

El tipo más popular es la de matriz de puntos por impacto. Como todas las impresoras modernas, el carro portapapel permanece quieto, mientras que la cabeza impresora se mueve a lo ancho del papel. La cabeza impresora contiene una columna de agujas verticales muy agrupadas, cada una de las cuales tiene un solenoide asociado. Cuando un solenoide se activa, la aguja correspondiente recibe un fuerte empujón, golpeando el papel a través de una cinta impregnada de tinta, por lo que hará un punto sobre el papel. Actuando sobre la



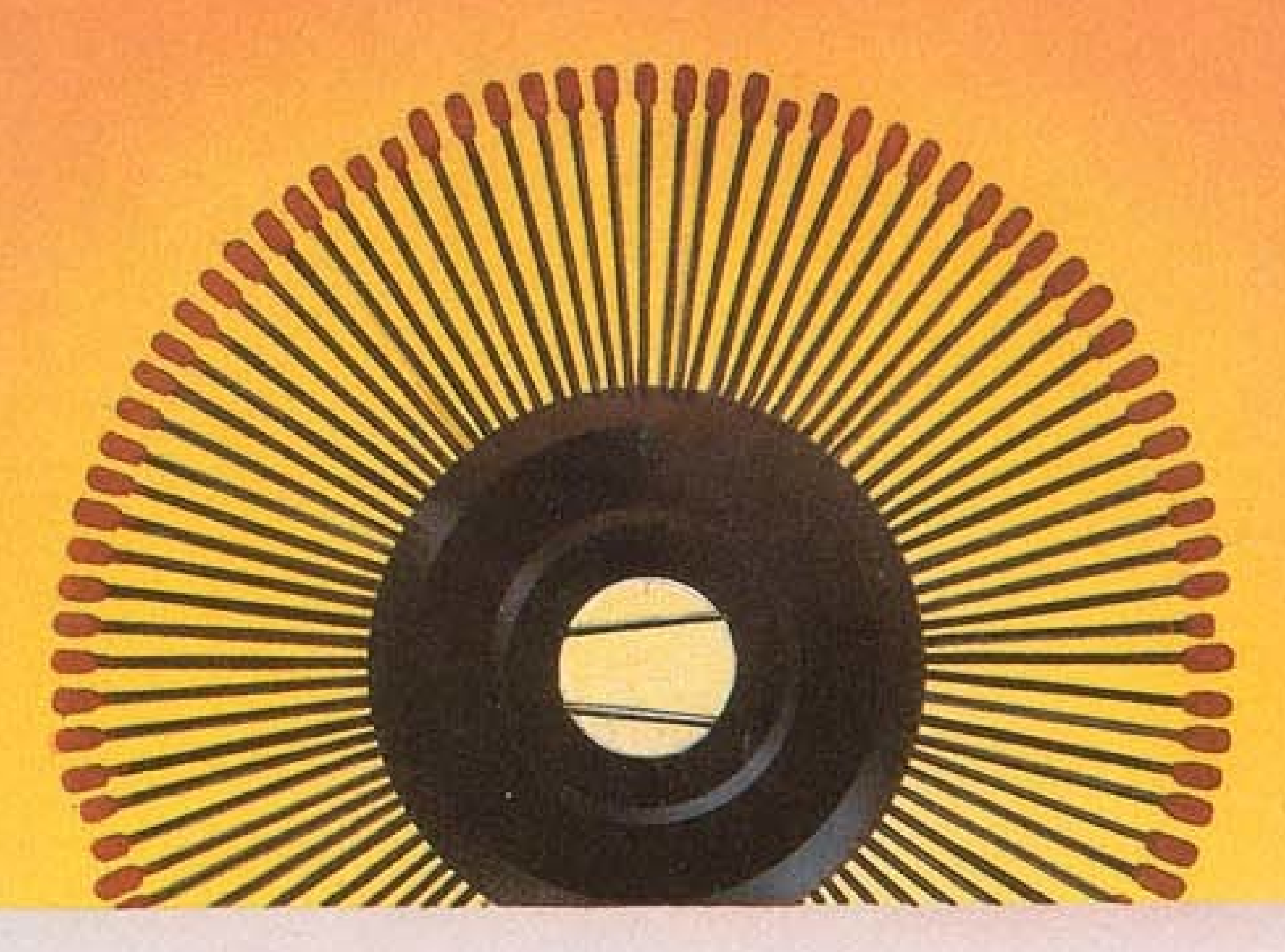
combinación de agujas correcta, y haciendo avanzar la posición de la cabeza de impresión, se formará un carácter.

Las velocidades de impresión de la impresora de matriz de puntos figuran entre las más rápidas disponibles, variando desde 80 caracteres por segundo (cps) a más de 400. Sin embargo, alguna de estas impresoras tiene la desventaja de que no presentan la «cola» de determinadas letras, tales como la «g» y la «q». Hacen que todo el carácter figure por encima de la línea, sin ninguna cola que asome por debajo. El poder presentar colas por debajo de la línea es deseable para tener claridad y una presentación atractiva, siendo una prestación disponible en las mejores impresoras de matriz de puntos.

Los primeros modelos de impresoras de matriz de puntos por impacto utilizaban una cabeza de impresión de siete agujas para producir caracteres en forma de matriz de 5 × 7 puntos, es decir, 5 columnas y 7 filas. Los modelos más recientes emplean nueve agujas para producir una matriz de 7 × 9 ó 9 × 9.

Incluso con la mejor impresora de matriz de puntos, los caracteres, aun siendo claramente legibles, aparecen como compuestos de puntos. Se adaptan muy bien para la mayoría del trabajo no profesional, y pueden imprimir dibujos a partir de la pantalla. Pero para una representación de calidad, se han hecho intentos de mejora de la claridad de impresión con la impresora de matriz de puntos.

Un método, llamado impresión multipaso, hace que la cabeza impresora imprima cada línea dos veces, una en una dirección, y una segunda vez en la dirección opuesta, todo ello sin hacer que el papel avance. En la segunda pasada, siempre hay, invariablemente, una pequeña falta de alineamiento entre las agujas impresoras y los caracteres ya impresos, por lo que las agujas tienden a imprimir en los espacios entre puntos. Se consiguen así efectivamente caracteres más marcados, menos punteados, pero, naturalmente, la velocidad de impresión queda reducida a la mitad.



Las impresoras de margarita son insuperables en cuanto a la calidad de la impresión. Para cambiar el conjunto de caracteres no hay más que cambiar la propia margarita.

Otro método (que no implica una importante pérdida en la velocidad de impresión) utiliza múltiples columnas de agujas en la cabeza de impresión. Los caracteres se van formando en pasos sucesivos: se activa primero la columna de la derecha, cuando vuelve a su posición de equilibrio, se activa la columna que está más a su izquierda, etc.

Aunque con estos métodos se consigue una mejora significativa, la calidad de la impresora de la matriz de puntos no es la mejor obtenible. Cuando se requiere una calidad elevada (como en la correspondencia formal), lo más adecuado es elegir una impresora de «fuente sólida». Fuente sólida quiere decir que las letras están formadas por tipos en relieve, que son una imagen especular exacta de la forma que se desea obtener, como ocurre con las letras de una máquina de escribir ordinaria.

IMPRESORAS DE FUENTE SOLIDA

Las primeras impresoras de fuente sólida que se usaron con los microordenadores eran adaptaciones de las máquinas de escribir eléctricas de bola. Aunque la calidad de impresión era excelente, la masa de la cabeza de impresión tipo pelota de golf (diseñada para una velocidad máxima de unos 15 caracteres por segundo, correspondiente a las mayores velocidades mecanográficas) retrasaba la impresión y la hacía poco fiable. En consecuencia, se fue desarrollando la impresora de margarita (conocida también como impresora de pétalos).

Las impresoras de plumilla encuentran gran aplicación en diseño con ayuda de ordenador, siendo muy útiles para generar una gran variedad de caracteres alfanuméricos.

Las impresoras de margarita llevan un elemento impresor intercambiable que se parece a una margarita con los pétalos surgiendo radialmente desde un núcleo central. Cada pétalo lleva un carácter en la punta, que es golpeado por un martillo cuando ha llegado al sitio correcto. Como la margarita es extraíble, es muy sencillo cambiar de un tipo de letra a otro; para ello no hay más que cambiar las ruedas. La calidad de impresión es generalmente igual, o mejor, que la de una máquina de escribir y muy superior a la de una impresora de matriz de puntos por impacto, si bien las velocidades de impresión son más lentas que las de las impresoras matriciales. Son normales velocidades entre 8 y 70 caracteres por segundo.

Para mejorar la velocidad de impresión, muchas modernas impresoras de margarita son bidireccionales, lo que quiere decir que imprimen una línea de izquierda a derecha y la siguiente de derecha a izquierda. En otras palabras, aprovechan el retorno de carro, que antes era un tiempo perdido, para escribir una línea. Estas impresoras disponen también de circuitos que dicen a la cabeza impresora cuándo es más rápido volver al final de la línea más corta para imprimir la siguiente línea, o continuar hasta el final del papel e imprimir la línea siguiente en dirección contraria.

Las impresoras de margarita permiten una serie de realces del texto. La negrita, el subrayado y las verdaderas colas son estándar en este tipo de impresoras. Se pueden combinar diferentes tipos de letra, cambiando las ruedas entre una primera y segunda pasada. Las modernas ruedas de margarita están hechas de plástico, por lo que son más ligeras. Los pétalos son elásticos, por lo que vuelven a su po-



sición de partida después de cada golpe. Los caracteres están hechos de una resina dura. Golpean la cinta con unos bordes muy afilados, de forma que la calidad de la impresión es muy buena y la cinta dura mucho tiempo. Hay tres tamaños de caracteres: cuatro, cinco y seis caracteres por centímetro. El coste de sustitución de una rueda es del orden de una copia en *cassette* de un buen juego de aventuras.

IMPRESORAS SIN IMPACTO

Las impresoras de matriz de puntos sin impacto no son tan populares. Las más conocidas son las electrostásticas y las térmicas. Las impresoras térmicas requieren utilizar un papel con un recubrimiento químico sensible al calor. El papel es caro, pero no hace falta cinta. La cabeza de impresión contiene una matriz de pequeños elementos calentadores. Una vez que se ha seleccionado la figura que ha de ser impresa, los elementos que la componen se calientan a una temperatura entre 100 y 150° C, oscureciendo los tintes termosensibles del papel.

Una de las impresoras más baratas del mercado es la propia impresora **ZX** de **Sinclair** que es de tipo térmico. Es compatible con el **Spectrum** y el **ZX81,** y se conecta a los terminales del port de usuario de la máquina. Es electrostática.

Naturalmente, con una suma tan modesta no vas a tener una impresora de la máxima calidad, pero te resultará satisfactoria para la mayoría de las necesidades que no requieren una impresión propia de cartas de calidad.

El papel se presenta enrollado y es muy estrecho, sólo mide 100 mm, distinguiéndose de los demás por su brillo plateado debido a su recubrimiento de aluminio.

Si tienes un **ZX81,** hay un gran inconveniente: necesitas una fuente e alimentación más potente para conectar la impresora, que se alimenta del ordenador.

Las impresoras electrostáticas requieren una aportación de *toner* líquido (normalmente partículas de carbón suspendidas en una solución de isoparafina). Los caracteres se forman cargando el papel cubierto y haciéndolo pasar por una solución de toner. Las partículas negras del *toner* se adhieren a las zonas eléctricamente cargadas del papel. El exceso de toner es eliminado y el papel se calienta para el fijado.

Las impresoras térmicas y electrostáticas son más fiables que las de matriz de puntos por impacto (porque tienen menos partes mecánicas) pero tienen una resolución mucho más baja.

Uno de los desarrollos recientes más interesantes es un registrador/impresora de plumilla de bajo costo. Sirve para dibujar (con plumillas con su propio depósito de tinta) caracteres preprogramados, si bien fue concebido específicamente para hacer dibujos. En los modelos de más bajo precio, se utilizan plumillas tipo bolígrafo para dibujar sobre un papel corriente.

Típicamente, la cabeza escritora contiene hasta cuatro plumillas, por lo que se pueden hacer gráficos multicolores. El conjunto de caracteres alfanuméricos normales resulta nítido y fácilmente legible, pero puedes programar tus propios caracteres utilizando las posibilidades de gráficos de tu ordenador. Las velocidades de impresión con el conjunto de caracteres normales son del orden de unos 12 caracteres por segundo. Un juego de cuatro plumillas de repuesto viene a costar aproximadamente lo mismo que un par de *cassettes* de datos.

CONSIDERACIONES SOBRE EL PAPEL

Cuando se elige una impresora lo más importante es saber qué aspectos quieres que tenga lo impreso. Esto se aplica no sólo al tipo de letra, sino también al tamaño del papel y a su apariencia. Además de los papeles con recubrimientos requeridos por las impresoras electrostáticas y térmicas, hay una amplia variedad de anchuras, que va desde 10 a 40 cm. Algunas impresoras pueden utilizar varios tipos y tamaños de papel; otras sólo uno. El papel se puede comprar en hojas sueltas (como el papel de escribir normal), en rollos o plegado. El papel plegado es el que normalmente se asocia con las copias de ordenador, lleva en cada borde una banda agujereada y se suministra como una hoja continua, con perforaciones en cada «página» para poder separarlo en hojas y cortarlo a la longitud conveniente.

Cada tipo de papel requiere su propio mecanismo de arrastre. Para el papel plegado se utiliza tracción por medio de ruedas dentadas. Los agujeros de los bordes del papel van engranando con los salientes de las ruedas de arrastre, que van girando para ir haciendo pasar el papel ante la cabeza impresora. Cuando se utiliza una impresora de impacto, se pueden sacar copias múltiples intercalando hojas de papel carbón.

Aparte de su bajísimo precio, la impresora **ZX** es notable por su pequeñez, cabe confortablemente en la palma de la mano y pesa menos de un kilogramo. Estas dimensiones limitan el número de caracteres por línea a 32. Igual que la pantalla del televisor.

Cuando el papel viene en hojas sueltas o en rollo, se requiere un mecanismo de arrastre por fricción (como el de las máquinas de escribir). El papel queda atrapado entre dos rodillos de presión. El papel en rollos se alimenta desde un eje; para las hojas

sueltas se requiere una bandeja de entrada independiente. Para la impresión de correspondencia de gran calidad de presentación, lo mejor es sin duda utilizar hojas sueltas; el papel plegado y en rollo están más indicados para listados de programas y gráficos.

INTERFACE

Independientemente del tipo de impresora que decidas comprar, el punto más importante a considerar es si es compatible, o puede hacerse compatible, con tu ordenador. El ordenador sólo puede comunicarse con la impresora si ambas máquinas están adecuadamente interconectadas.

Un *interface* consiste esencialmente en el *hardware* (junto con el posible *software* requerido para controlarlo) que hace posible conectar dos sistemas. Algunos sistemas se conectan simplemente con un enchufe, porque ya se han hecho compatibles durante su fabricación o porque se venden con *interfaces.* Otros se venden sin nada de *interface,* o con sólo una parte de *interface,* por ejemplo, sólo con el *hardware.*

No esperes a haber comprado la impresora para averiguar las exigencias del *interface,* porque te puedes encontrar con que el coste del *interface* asciende a un tercio del coste de la máquina.

Si ambas máquinas proceden del mismo fabricante, lo normal es que ya sean compatibles, o puedan serlo añadiéndoles un *interface.* Y los fabricantes independientes de impresoras intentan hacer sus productos compatibles con el mayor número de ordenadores posible, normalmente suministrando los *interfaces* adecuados. Por eso, aunque el *interface* es crucial, no tiene por qué ser un problema, si te aseguras bien antes de comprar.

Se pueden comprar módulos de conversión de *interface* para algunas máquinas, pero también puede ocurrir que el *software* que quieras utilizar —especialmente algunos programas procesadores de textos— requiera su *interface* particular.

Cuando hayas decidido el tipo de impresora que vas a comprar, tienes que tener una idea del precio, *interface* incluido. Naturalmente, puede que el precio haya sido desde el principio el factor decisivo. Ten cuidado con algunas máquinas que se venden sin cable de interconexión, que puede llegar a costarte el doble que muchos juegos de disco de bajo precio.

Tanto si eliges una impresora que cueste tanto como tu ordenador, como si cuesta diez veces más, te recomendamos que examines el paquete de *software* y la impresora que quieres ver trabajando junto con tu ordenador, antes de quedarte solo. Normalmente hay disponibles unidades de demostración, pero a lo mejor no están a la vista, por lo que no dudes en preguntar y pedir aclaraciones.



JUEGOS PARA LA MENTE

MIND GAMES ESPAÑA, S.A.

SONIMAG-85
nivel 10
stand n.º 5



ORIGINAL

SPECTRUM 48K

P.V.P.
2.495 ptas.

C−64

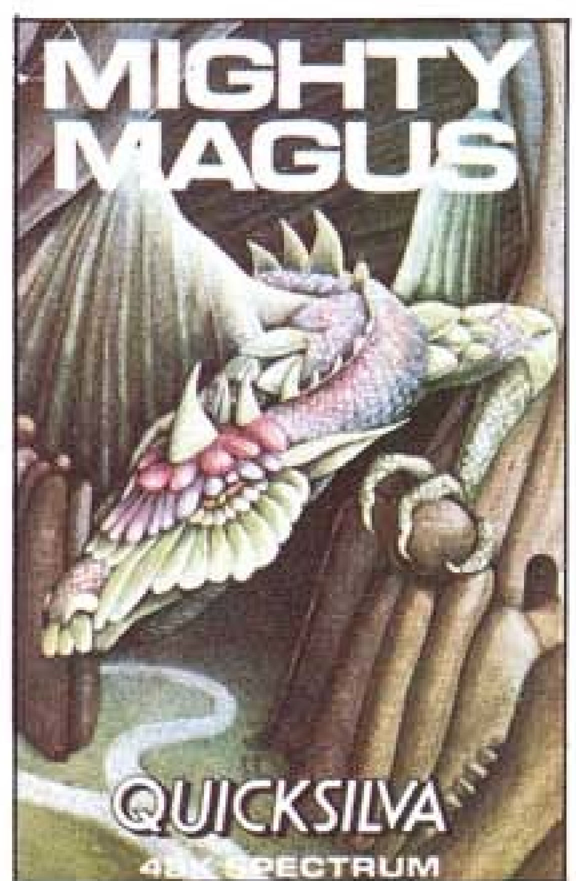
GRAN NOVEDAD

SPECTRUM 48K

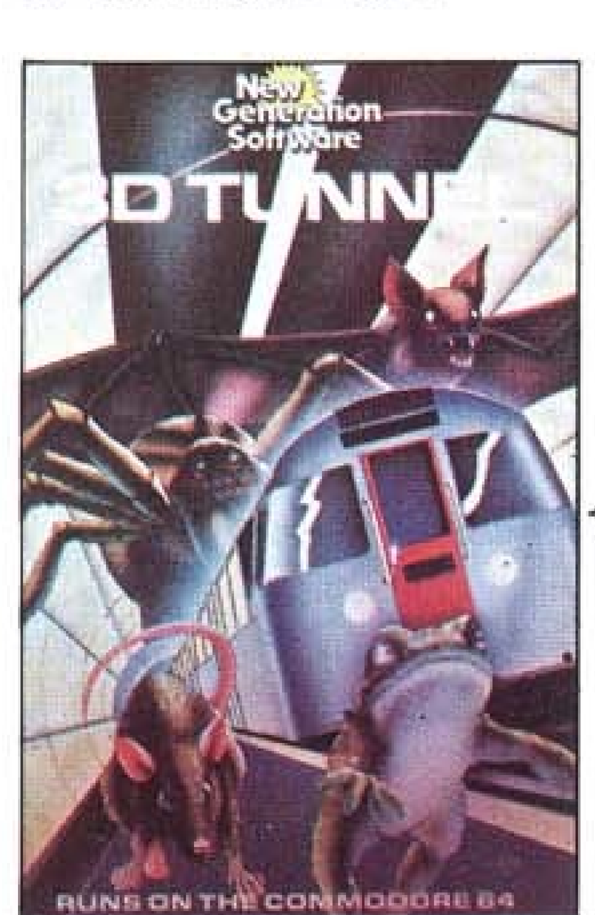
FANTASTIC VOYAGE
QUICKSILVA
48K SPECTRUM

SPECTRUM 48K

P.V.P.
2.495 ptas.



SPECTRUM 48K

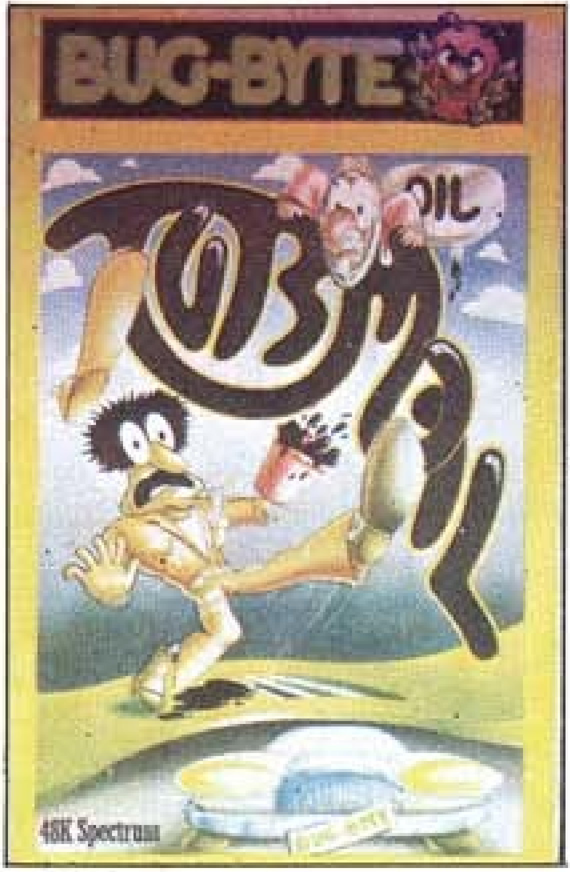


CBM 64

P.V.P.
1. 495 ptas.



CBM 64



MSX

P.V.P.
2.495 ptas.



MSX

Para más información dirigirse a:

EDITADO Y DISTRIBUIDO POR:

MIND GAMES ESPAÑA, S.A.

Mariano Cubí, 4 Entlo. Tel. 218 34 00 - 08006 Barcelona

También en venta en los mejores Establecimientos

Programación

NO LO OLVIDES, ARCHIVALO

- VENTAJAS DE ALMACENAR DATOS EN UN PERIFERICO
- LA MARCA FIN DE FICHERO
- TERMINADORES MUDOS
- USO DE LOS DATOS

Abre un canal hacia tu *cassette* o unidad de discos y envía por él tus datos importantes. Saber manejar ficheros con soltura es una de las claves de la programación.

Todo programa que maneje gran cantidad de datos ha de contar con alguna forma conveniente de almacenarlos. Existen cortos programas que permiten almacenar un listín telefónico sencillo en sentencias de tipo DATA. El programa busca e imprime un dato determinado, pero no tiene manera de actualizar la lista sin interrumpir (con BREAK) el programa y alterar las líneas del mismo. Además solamente se puede almacenar la información dentro del propio programa. De esta manera está uno encadenado a la información que contenga, lo cual no es muy satisfactorio.

Evidentemente, es mucho mejor almacenar los datos separadamente en una cinta o disco, en lugar de mantenerlos dentro del programa. De esta forma, el ordenador puede leer la información con la misma facilidad con que lee las líneas del programa, pero se pueden modificar los datos con mucha mayor facilidad, y, además, el usuario del programa no tiene que ser un experto en programación para modificar dichos datos.

En una cinta o en un disco, los datos se almacenan en un fichero, y muchos de los programas de aplicaciones almacenan los datos de esta forma. Así el fichero de *hobbies,* finanzas caseras, calendario y hoja electrónica, que iremos publicando, son algunos de los programas que han de poder manejar grandes cantidades de información. Si examinas atentamente estos programas, identificarás la parte que almacena y carga los datos. En todos los casos es una pequeña parte del programa, siendo mucho más lo que se dedica a asuntos tales como la disposición de memoria, mensajes al usuario *(prompts),* visualizaciones y cosas por el estilo.

Almacenando la información separadamente del "núcleo" del programa, tienes lo que se llama una memoria virtual: el programa básico se puede usar con una variedad de ficheros diferentes, cada uno con un número distinto de registros o campos, con una disposición adecuada al objeto de cada fichero.

El almacenamiento secuencial de datos es uno de los métodos preferidos. Su programación es muy sencilla y los principios en que se basa son fáciles de entender. Además, los datos se pueden convertir a un formato que realmente permite pasarlos de un programa a otro o, incluso, de un ordenador a otro.

Un fichero secuencial difiere de los ficheros de otra clase en que almacena los datos en serie, siendo cada dato separado del siguiente por un solo byte. Toda la información se recoge en la memoria antes de ser transferida a un fichero para su almacenamiento.

El primer paso para guardar un fichero secuencial es introducir la información en tu dispositivo de almacenamiento. Esto se hace con el comando OPEN. Los comandos de uso más corriente se describen en los manuales del usuario.

Este artículo explica con detalle la forma de crear tus propios ficheros de modo que puedas incorporarlos en tus programas.

CREACION DE UN FICHERO

El programa muestra cómo realizar una versión más sofisticada del directorio telefónico mencionado anteriormente. Pero no tendrás dificultad en cambiar los mensajes de pantalla y cadenas alfanuméricas para adaptarlos a otros tipos de información.

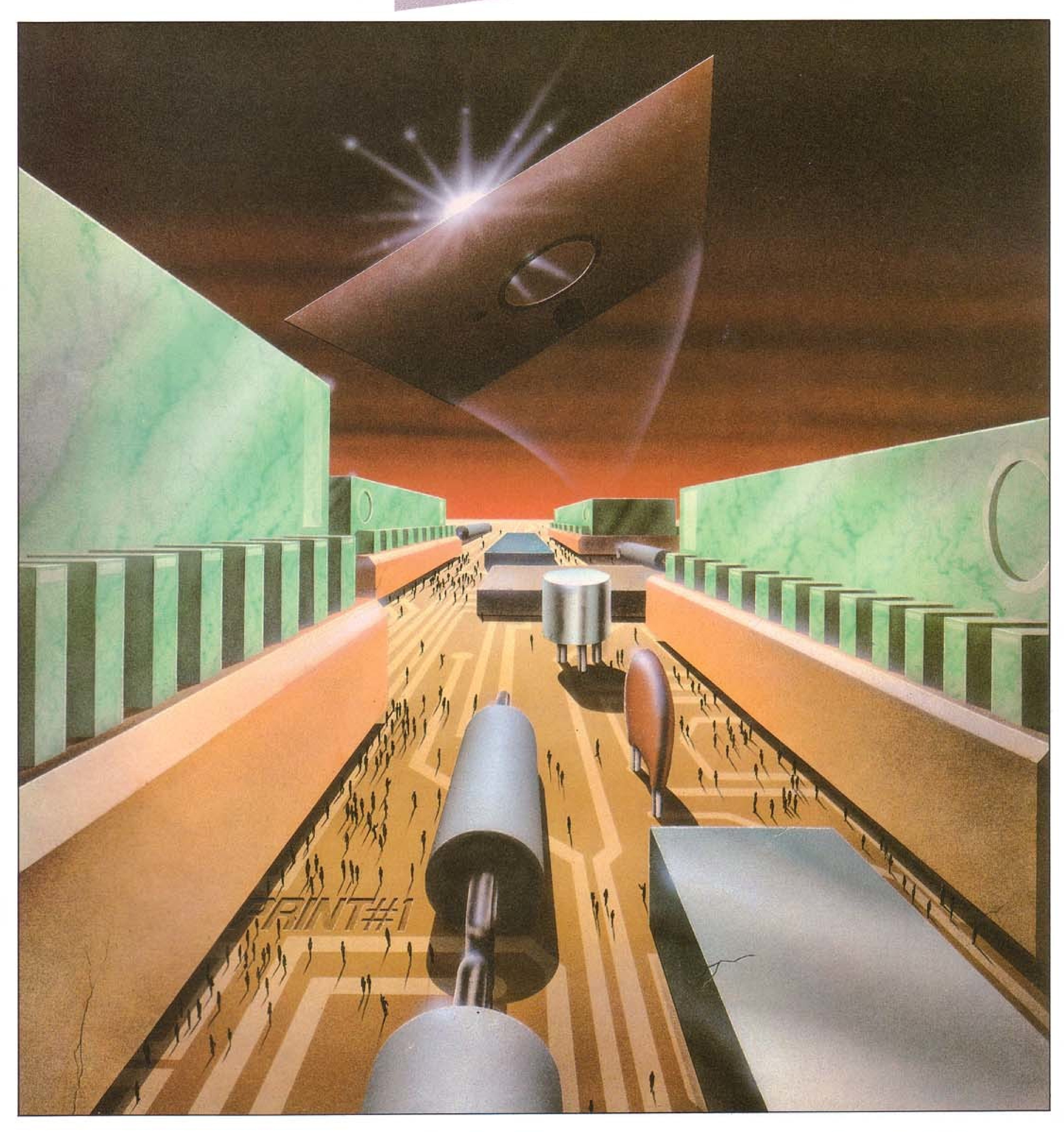
Las entradas admitidas son nombre, apellido y número. Como los números de teléfono contienen espacios y posiblemente también guiones, es mejor tratarlos como una matriz unidimensional. Con otras clases de datos tal vez prefieras tener algunas cadenas de matriz y algunas cadenas numéricas. No importa el tipo de variable que se almacena, o en qué orden, ya que se vuelven a leer en el mismo orden y con el mismo tipo de variable.

Se puede enviar información directamente a un fichero, pero es más corriente y más conveniente almacenarla primero en una cadena, almacenando posteriormente toda ella.

La primera parte del programa establece un sencillo bucle que te permitirá introducir los datos. Puedes meter todos los nombres y números que quieras, hasta el número dimensionado en la línea 10. Si quieres meter más de 50 tienes que cambiar la dimensión. Cuando termines, en vez de seguir meteniendo un nombre y número, pulsa ENTER y la subrutina de entrada se detendrá.

La segunda mitad del programa, desde la línea 100 en adelante, escribe los datos en el fichero. Esta parte se explica después de cada programa.

```
10 DIM A$(50,15): DIM
   B$(50,15): DIM T$(50,12):
   DIM N(1)
20 LET N=0
30 LET N=N+1
40 INPUT "NOMBRE";A$(N)
50 INPUT "APELLIDO";B$(N)
60 INPUT "NUMERO DE
   TELEFONO";T$(N)
70 IF A$(N)<>"□□□□□□□□
   □□□□□□□" AND N£50 THEN
   GO TO 30
80 CLS : PRINT "ALMACENANDO
   LOS DATOS"
```



Utiliza esta sección si guardas los datos en cinta:

```
100 SAVE "CONTADOR" DATA N()
110 SAVE "NOMBRES" DATA A$()
120 SAVE "APELLIDOS" DATA
    B$()
130 SAVE "TELEFONOS" DATA T$()
140 PRINT "DATOS
    ALMACENADOS"
150 STOP
```

Utiliza esta sección si tienes un Microdrive.

```
90 INPUT "ESCRIBE EL NUMERO
    DEL DRIVE □";DRV
100 SAVE *"M";DRV;
    "CONTADOR" DATA N()
110 SAVA *"M";DRV;"NOMBRES"
    DATA A$()
120 SAVE *"M";DRV;
    "APELLIDOS" DATA B$()
130 SAVE *"M";DRV;
```

```
    "TELEFONOS" DATA T$()
140 PRINT "DATOS
    ALMACENADOS"
150 STOP
```

La primera sección del programa, desde la línea 10 a la 80 es común para la cinta y el **Microdrive;** sin embargo, la segunda parte, que es la que almacena los datos, es diferente. Ambas versiones almacenan los datos en forma de cadena, pero con el **Microdrive** tienes que especificar qué unidad se está usando (puede haber más de una) y añadir "M"; DRV; después de cada comando SAVE. No olvides que hay que asignarle a cada cadena un nombre de fichero antes de almacenarla, y que antes de cada cadena hay que poner una instrucción DATA.

Introduce ahora la versión apropiada del programa, y a continuación prueba a meter algunos datos y almacenarlos, quedando listos para el próximo programa.

Hay otra manera de almacenar los datos cuando se utiliza un **Microdrive.** El método anterior tiene el inconveniente de que hay que guardar toda la cadena, aunque sólo se utilice una parte de la misma. Esto consume un espacio precioso de la cinta del **Microdrive.** La versión opcional que sigue a continuación, almacena cada elemento de la cadena separadamente y se detiene cuando se acaban los datos.

```
100 INPUT "ESCRIBE EL NUMERO
    DEL DRIVE□";DRV
110 OPEN #4;"M";DRV;
    "FICHERO"
120 PRINT #4;N
125 FOR L=1 TO N
130 PRINT #4;A$(L)'B$(L)'
    T$(L)
140 NEXT L
150 CLOSE #4
160 PRINT "DATOS ALMACENADOS"
170 STOP
```

Igual que antes, has de teclear primero el número de la unidad. Seguidamente, la línea 110 abre un canal de comunicaciones con el **Microdrive.** Puedes utilizar cualquier canal del 4 al 15 (los canales 0 al 3 se reservan para la pantalla y la impresora de **ZX,** siendo el 4 el que se usa más frecuentemente. A continuación, la línea 120 escribe el valor de N, el número de datos diferentes, que pasarán al fichero por este canal, y la línea 130 escribe cada grupo de datos, uno cada vez, hasta alcanzar un número máximo de N. Seguidamente se cierra el canal con CLOSE#4.

Es muy importante teclear estas líneas exactamente como se muestran. En las líneas 110 y 120 hay que utilizar el punto y coma para separar los grupos de datos, y en la línea 130 hay que poner un punto y coma después de PRINT#4, separando el resto de las variables con apóstrofes.

Si quieres probar este programa, guarda primero la otra versión (con SAVE), teclea las nuevas líneas y vuelve a intentar almacenar datos, preferiblemente en una cinta distinta para que no se mezclen. Los datos están todavía almacenados en las cadenas. ¡Así que ten cuidado de no teclear NEW!

LECTURA DE UN FICHERO

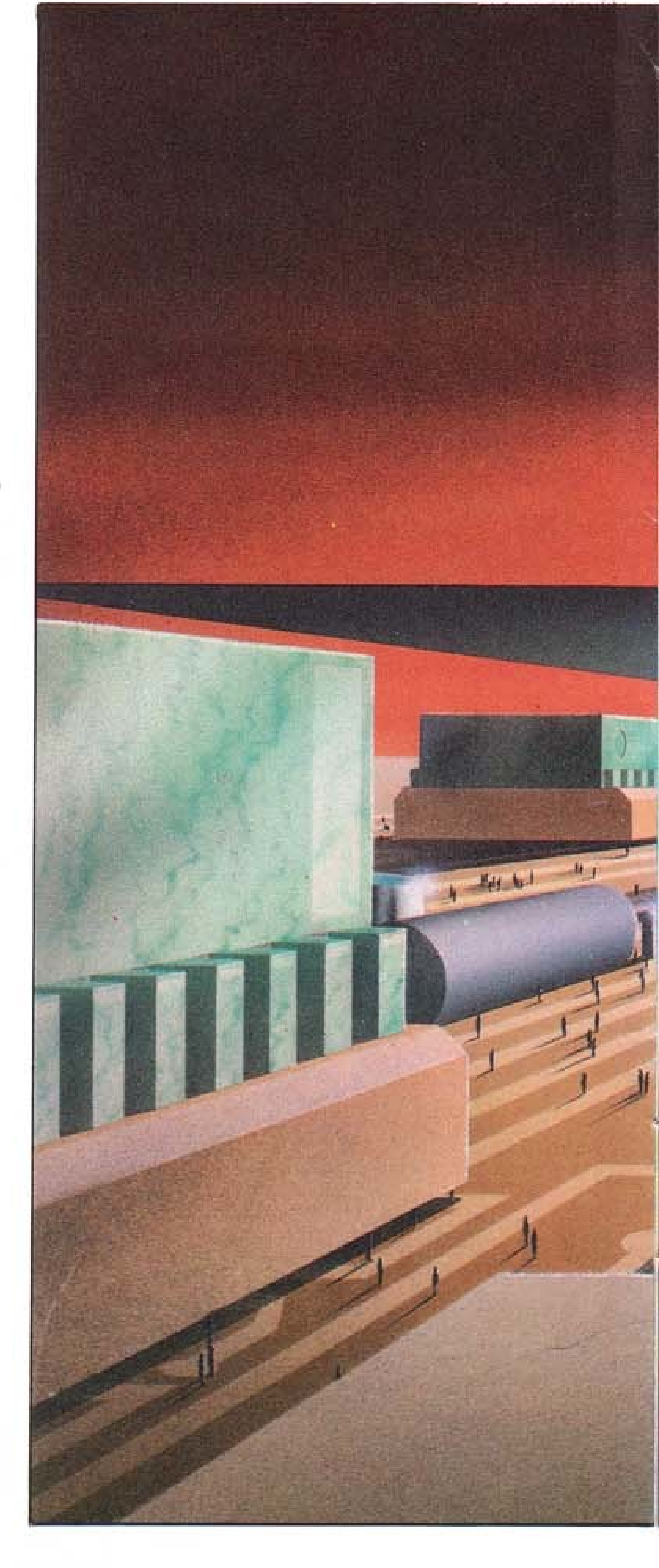
El programa para leer el contenido de un fichero, y trasladarlo a la memoria, es el inverso del programa de escritura. También se dedica una línea para presentar la información en pantalla a fin de que puedas comprobar que realmente han sido leídos los datos.

Utiliza esta sección para la cinta:

```
200 CLEAR
210 DIM N(1): LOAD
    "CONTADOR" DATA N()
215 DIM A$(N(1),15): DIM
    B$(N(1),15): DIM
    T$(N(1),12)
220 LOAD "NOMBRES" DATA A$()
230 LOAD "APELLIDOS" DATA
    B$()
240 LOAD "TELEFONOS" DATA
    T$()
250 FOR L=1 TO N(1)
260 PRINT A$(L),B$(L),T$(L)
270 NEXT L
```

Para el **Microdrive** utiliza:

```
200 CLEAR
205 INPUT "ESCRIBE EL NUMERO
    DEL DRIVE□";DRV
210 DIM N(1): LOAD *"M";DRV;
    "CONTADOR" DATA N()
215 DIM A$(N(1),15): DIM
    B$(N(1),15): DIM
    T$(N(1),12)
220 LOAD *"M";DRV;"NOMBRES"
    DATA A$()
230 LOAD *"M";DRV;
    "APELLIDOS" DATA B$()
240 LOAD *"M";DRV;
    "TELEFONOS" DATA T$()
250 FOR L=1 TO N(2)
260 PRINT A$(L),B$(L),T$(L)
270 NEXT L
```



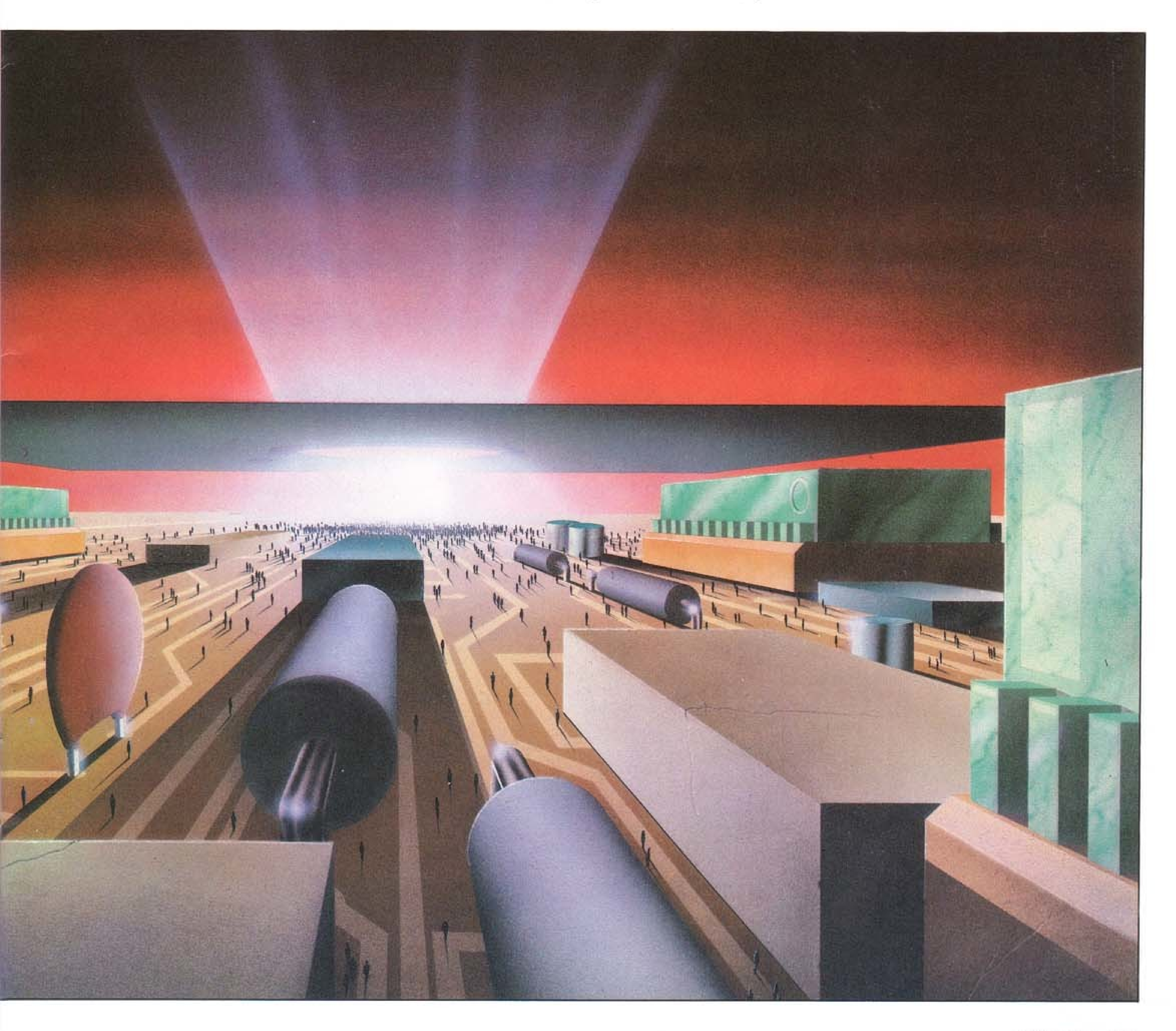
Para utilizar este programa rebobina la cinta hasta el principio del fichero y a continuación pulsa RUN 200 para leer los datos; no vuelvas a ejecutar (RUN) otra vez el programa completo.

La línea 200 se ha puesto para borrar los nombres y números que pueda haber en la memoria del ordenador, de forma que el programa sólo presente los que lea del fichero. La línea 210 dimensiona una nueva cadena N() de un elemento. Se utiliza para almacenar el valor de N que se guardó la última vez. En la siguiente línea se dimensionan las otras cadenas, a partir de este valor. Como ahora se conoce el número máximo de elementos, se ahorra la lectura de los elementos vacíos de la cadena.

El resto del programa es muy sencillo, y se puede escribir simplemente, sustituyendo SAVE por LOAD. No es necesario cerrar un fichero que se ha abierto para lectura, sólo es necesario en escritura.

De nuevo aquí hay una versión opcional para el **Microdrive** (no intentes probar todas estas versiones mezcladas). Y también se parece al procedimiento de escritura, excepto en vez de INPUT#4, se pone PRINT#4, y que en la línea 230, las variables van separadas por punto y coma.

```
200 CLEAR
205 INPUT "ESCRIBE EL NUMERO
    DEL DRIVE";DRV: OPEN
    #4;"M";DRV;"FICHERO"
210 INPUT #4;N: DIM
    A$(N,15): DIM B$(N,15):
    DIM T$(N,12)
220 FOR L=1 TO N
230 INPUT #4;A$(L);B$(L)
    ;T$(L)
235 PRINT A$(L);B$(L);T$(L)
240 NEXT L
250 CLOSE #4
```



EL FINAL DEL FICHERO

En el último programa se utilizaba la variable N en el bucle que controla el procedimiento de lectura, pero no siempre es posible saber de antemano cuántos datos hay que leer. Una manera de obviar este inconveniente sería utilizar una marca de fin de fichero. Desgraciadamente, el **Spectrum** no utiliza marca de fin de fichero (pero podrás poner tu propia marca muda, como veremos después).

He aquí el programa completo a partir de la línea 100 para que puedas compararlo con la versión anterior. Guarda el programa viejo (con SAVE), cambia las líneas que siguen e intenta nuevamente leer y escribir datos. Puedes quedarte con la versión que te vaya mejor.

Otra manera de detectar la llegada del final de un fichero es tener algún tipo de variable "muda". De hecho, ésta es la única forma de detectar un final del fichero en un **Microdrive** en un fichero de cinta, en el **Spectrum,** no hay ninguna manera de hacerlo. El procedimiento es muy simple. En la rutina original de entrada, cuando se empiezan a meter los datos en el ordenador, se puede introducir una palabra o valor mudo, detrás del último dato correcto. Podría ser la palabra FIN, o ZZZ, ó 999 o cualquier cosa que normalmente no vayas a encontrate en el fichero real. Para ver cómo se utiliza en la práctica la variable muda, modifica las siguientes líneas. Cuando termines de meter los datos esta vez, teclea FIN, en lugar del siguiente NOMBRE, después de ENTER para los otros dos datos pedidos en la pantalla.

Sólo se puede hacer con la versión **Microdrive,** que almacena un dato cada vez. Hay que modificar la línea 70 como se indica, a continuación se dan las rutinas completas de escritura y lectura.

```
 70 IF A$(N)<>"FIN" AND N<50
    THEN GO TO 30
100 INPUT "ESCRIBE EL NUMERO
    DEL DRIVE□";DRV
110 OPEN #4;"M";DRV;
    "FICHERO"
120 LET L=0
125 LET L=L+1
130 PRINT #4;A$(L)'B$(L)
    'T$(L)
140 IF A$(L)<>"FIN" AND L<50
    THEN GO TO 125
150 CLOSE #4
160 PRINT "DATOS ALMACENADOS"
170 STOP
200 CLEAR : DIM A$(50,15):
    DIM N$(50,15): DIM
    T$(50,12)
205 INPUT "ESCRIBE EL NUMERO
    DEL DRIVE□":DRV
210 OPEN #4;"M";DRV;
    "FICHERO"
215 LET L=0
220 LET L=L+1
230 INPUT #4;A$(L);B$(L)
    ;T$(L)
235 PRINT A$(L);B$(L);T$(L)
240 IF A$(L)<>"FIN" AND L<50
    THEN GO TO 220
250 CLOSE #4
```


SOFTWARE PARA EL FUTURO
Century City
SENTINELS
Software for the Future.
Attack on Atlantis
Software for the Future.
Sea Quest
Vic 20 + 16K
The HOUSE on DAMNED HILL
Software for the Future.
Olympic Challenge
Software for the Future.
London to Paris
Air Race
Vic 20 + 16K.
ACCELERATOR
Software for the Future.
LASER LORD
Software for the Future.
MAN TRAP
Software for the Future.
P.V.P.
795 Ptas.
SONIMAG 85
nivel 10
stand nº 5
si deseas más información,
dirígete a tu establecimiento habitual o:
MGE SOFTWARE
MIND GAMES ESPAÑA, S.A.
Mariano Cubi, 4 Entlo. Tel. 218 34 00 - 08006 Barcelona

QUE ES LA HOJA DE TRABAJO

Si tienes que manejar muchas cifras, es un buen momento para hacer que te ayude tu micro. Probablemente es mucho mejor que tú con los números.

Una penosa tarea, que casi todo el mundo se ve obligado a realizar, es el control de los gastos propios, lo que constituye una buena idea de aplicación para el usuario de un micro, que puede servirse del mismo para saber a dónde va a parar su dinero. Pero ahora nos vamos a referir a un sistema distinto, modelado a partir de otro que utilizan los contables profesionales: la hoja de trabajo (también llamada hoja electrónica).

Las hojas de trabajo figuran entre los programas más versátiles, con unas posibilidades casi ilimitadas para el manejo de información numérica. Además, no están en absoluto restringidos al uso de datos financieros.

Este artículo consta de tres partes. Para empezar daremos una visión de lo que puede hacer una hoja de trabajo y para qué se utiliza. Seguidamente podrás programar tu propia y sencilla hoja, utilizando el listado que empieza en la segunda parte de este artículo. En la última parte te daremos instrucciones detalladas que harán por ti una parte del trabajo.

QUE ES UNA HOJA DE TRABAJO

Las hojas de trabajo utilizan una de las mayores ventajas que tienen los ordenadores: su capacidad de hacer cálculos muy rápidamente. En esencia hasta el ordenador más grande no es más que una máquina de sumar. De hecho, lo único que maneja un ordenador son números, y quienes se han sumergido en el código máquina refrendarán esta afirmación.

La hoja de trabajo puede ser una herramienta enormemente potente. Se utiliza normalmente para realizar cálculos financieros, pero también puede servir para construir toda clase de modelos de cálculo. Sustituye a los viejos métodos a base de lápiz, papel y calculadora, utilizados por los contables para calcular los beneficios de una empresa. A nivel doméstico la hoja de trabajo puede servir para el seguimiento de los gastos personales, o para tratar los datos relacionados con un *hobby*.

La hoja de datos usada tradicionalmente en contabilidad, en la que por ejemplo se registran los ingresos y los gastos, consiste en una gran hoja de papel, que normalmente ocupa la extensión de un par de páginas. Se divide horizontalmente en filas y verticalmente en columnas. Se produce así una retícula de cajas, celdillas o «células». En la fila superior el contable escribe, por ejemplo, los nombres de los meses del año, de forma que cada columna se refiere a un mes. Junto a la cuadrícula se disponen encabezamientos, tales como ingresos y gastos. Para un análisis más detallado puede introducir subtítulos tales como ventas propias, exportaciones, costes laborales, costes materiales brutos, etc. De este modo cada fila está relacionada con un área específica de ingresos y gastos.

El título final en el margen de la cuadrícula es normalmente pérdidas/ganancias, y las cifras que figuran al final de cada columna muestran el beneficio o pérdida habidos cada mes. Al final de cada fila, en la columna número 13, figura el ingreso o gasto total en cada área espcífica al cabo del año.

Rellenar las celdillas con los números es una tarea laboriosa, se disponga o no de un ordenador. Pero los contables que utilizan una hoja de trabajo de papel, han de abordar además la pesada tarea de calcular las cifras de pérdidas o ganancias. Esto supone la suma de todas las cifras de ingresos, la de todas las cifras de gastos y posteriormente restar el total de gastos del total de ingresos.

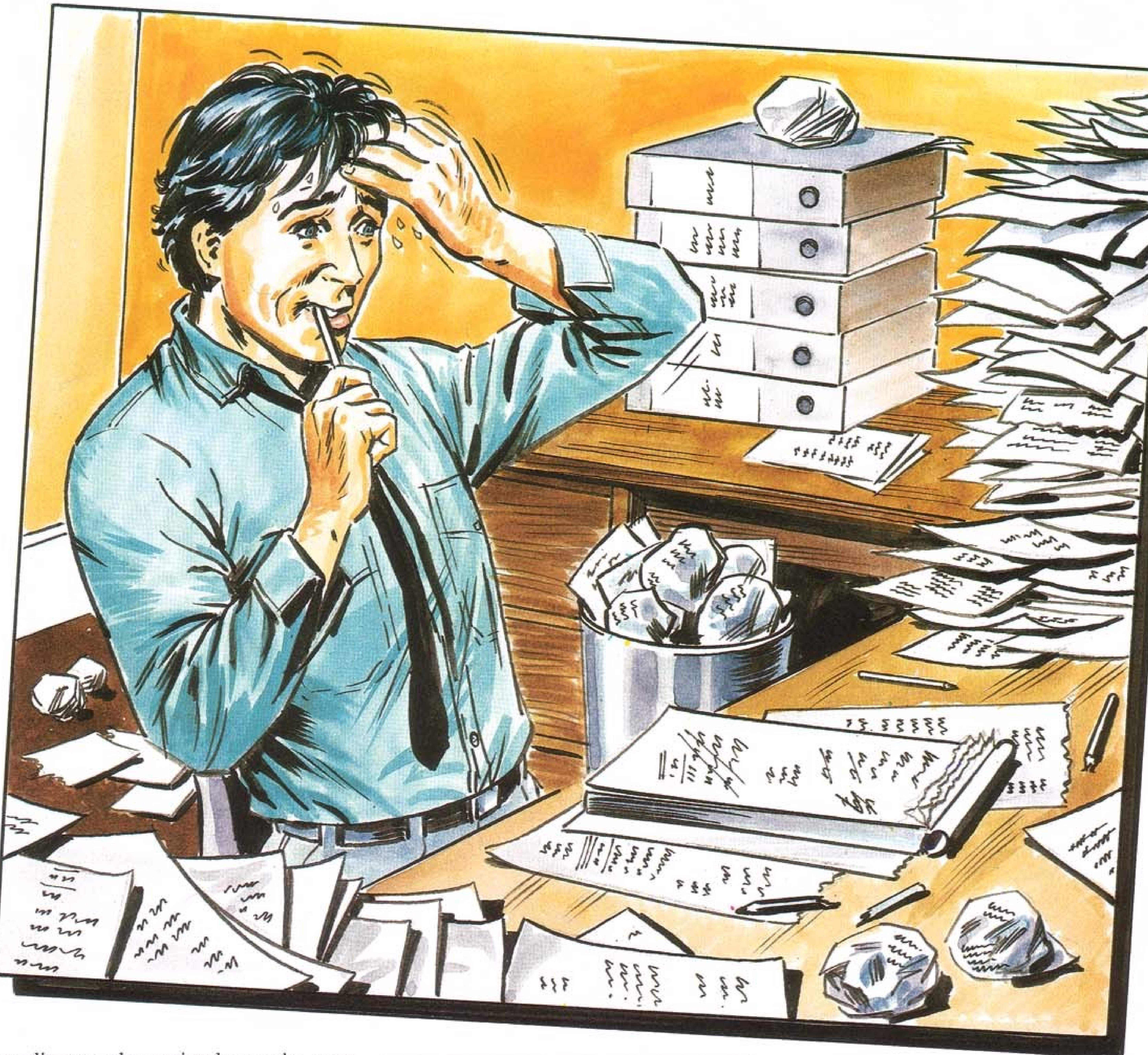
ENTRA EL ORDENADOR

En muchos aspectos, la hoja de trabajo de un ordenador es igual que la que utiliza papel, con la misma cuadrícula, dividida en filas y columnas. En la práctica, para que las casillas tengan un tamaño razonable, sólo se representa en pantalla una pequeña sección de la hoja; de esta manera la pantalla actúa como una «ventana» que permite observar la zona particular en que estés interesado.

Aquí también, igual que en la versión de papel, puedes meter lo que quieras en las casillas vacías, que no tienen significado alguno hasta que no las definas. Puedes introducir una etiqueta o título, o números, dependiendo de lo que quieras que presente la hoja de trabajo.

Hasta aquí, la hoja en el ordenador es, si acaso, un poco más complicada que una hoja de papel. Pero su verdadera potencia reside en la capacidad de procesar la información que introduces en ella. Por debajo de la hoja en blanco, que tú vas rellenando, existe otra hoja escondida. Esta segunda hoja de trabajo es la que dice al ordenador lo que debe ir haciendo con la información que encuentra en cada casilla. Sin embargo, dicha hoja de trabajo «escondida» no tiene nada de misterioso, ya que eres tú quien la pones allí, y siempre está disponible para que puedas examinarla o modificarla en cualquier momento.

Volviendo a nuestro esforzado contable, supongamos que quiere representar en una segunda columna el tanto por ciento de impuestos corres-



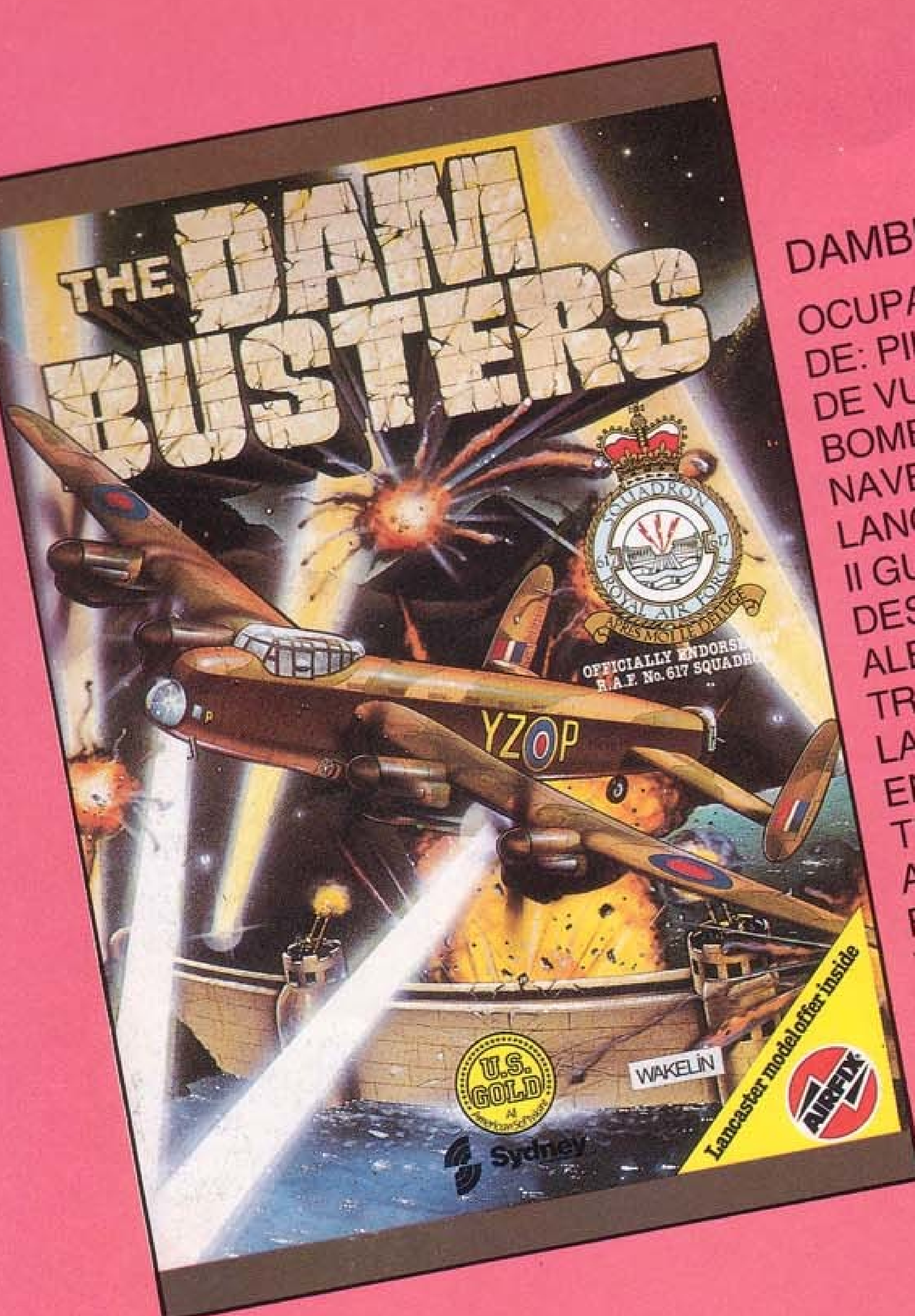
pondiente a determinado precio, y en otra columna la suma de las dos cantidades anteriores. Con el ordenador, puede hacer un programa que haga todo esto a su gusto. Todo lo que tiene que hacer es introducir una instrucción en cada una de las celdillas de la columna dos, que mande al ordenador multiplicar el número de la columna uno por un porcentaje fijo. Una instrucción análoga en cada una de las casillas de la columna tres, hará que el ordenador calcule el total requerido, sumando los contenidos de las correspondientes casillas de las dos columnas anteriores.

PREDICIENDO EL FUTURO

Otro problema con el que se enfrenta el contable con su gran hoja de papel, es la introducción de modificaciones. Por ejemplo, un aumento de los costes laborales supone volver a calcular todas las cifras de gastos y restarlas otra vez de las cifras de ingresos, para encontrar el valor actualizado de pérdidas y ganancias. Si solamente estás anotando cifras, la cosa puede no ser muy grave, pero si estás haciendo un pronóstico a un año vista, o aún más lejos, podría significar centenares de nuevos cálculos. Esta es la típica tarea que, aunque se realice con ayuda de una calculadora, requiere mucho tiempo, aburre a cualquiera y está muy sujeta a errores. Una labor de este tipo la hace un ordenador en unos milisegundos.

Si introduces correctamente los números en la hoja de trabajo —cosa no siempre tan sencilla como pudiera parecer—, cualquier cambio en una cifra producirá automáticamente los ajustes adecuados en todas las cifras relacionadas con ella.

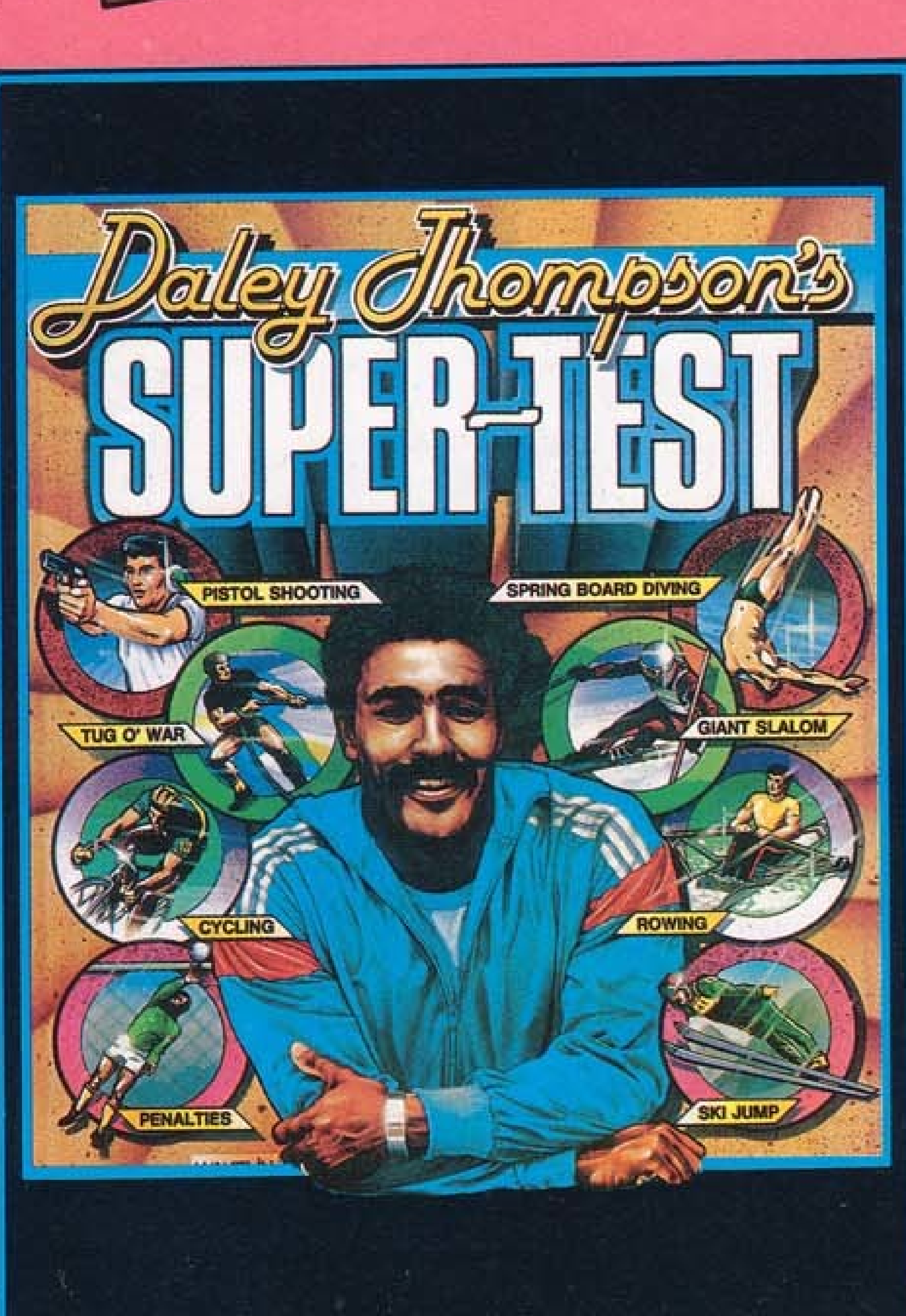
Si, por ejemplo, se cambia la cifra



DAMBUSTERS

OCUPA LOS PUESTOS DE: PILOTO, INGENIERO DE VUELO, ARTILLERO, BOMBARDERO Y NAVEGANTE DEL LANCASTER QUE EN LA II GUERRA MUNDIAL DESTRUYO LAS PRESAS ALEMANAS. VOLARAS A TRAVES DEL CANAL DE LA MANCHA Y EUROPA ENFRENTANDOTE A LOS TEMIBLES ME-110 ALEMANES, ZEPPELINES, FOCOS ANTIAEREOS Y TODOS LOS DEMAS PELIGROS A LOS QUE SE ENFRENTO EL COMANDO INGLES.

SPECTRUM/COMMODORE



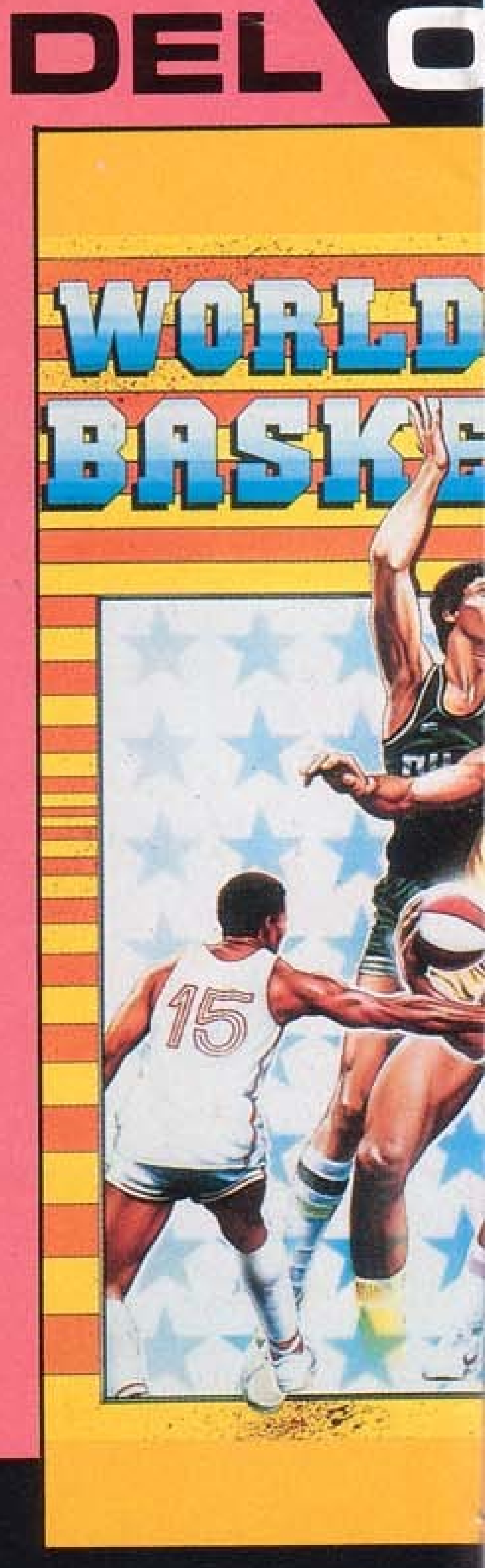
SUPERTEST

LA CONTINUACION DEL DECATHLON, EL MAS COMPLETO DE LOS JUEGOS DEPORTIVOS. CONSIGUE SER MEDALLA DE ORO EN: TIRO DE PISTOLA, SALTO DE TRAMPOLIN, PENALTIES, CICLISMO, SLALOM GIGANTE, REMO, TIRO DE CUERDA Y SALTO CON SKI. HECHO POR LOS MISMOS PROGRAMADORES DEL DECATHLON E HYPERSPORTS.

SPECTRUM

Soft
Pres

LOS E
DEL C



BASKETBALL

NO TE CONFUN
AUTENTICO JUEG
QUE HAS VISTO E
DE MEJORES GRAF
Y ADEMAS ¡GF
PROGRAMA UNA M
DE R
HUYE DE LAS

SPE

DISTRIBUCION EXCLUSIVA PARA ESPAÑA ERBE SOFTW

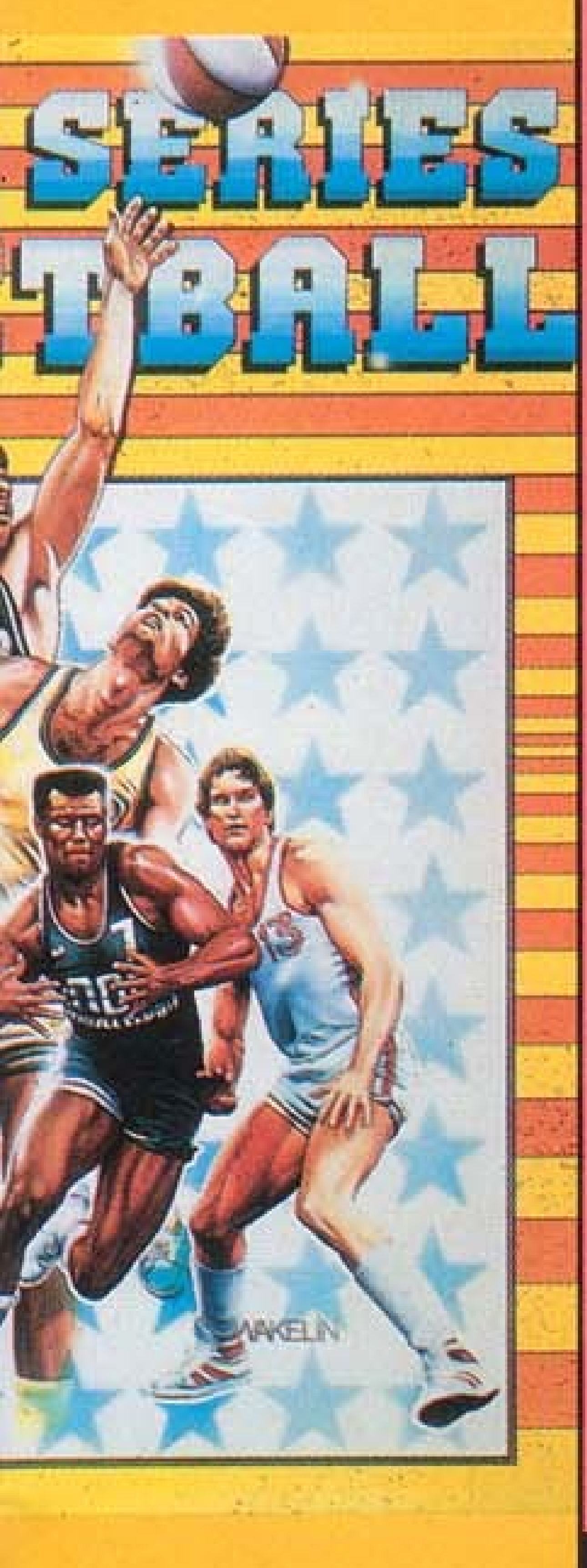
tware
senta

XITOS
TOÑO

FRANKIE GOES TO HOLLYWOOD

EL EXITO DEL MOMENTO. ALGO MAS QUE UN JUEGO. INCONTABLES PANTALLAS EN LAS QUE TODO PUEDE SUCEDER. DESDE TENER QUE DESCUBRIR AL ASESINO DE UN MISTERIOSO CRIMEN HASTA ENFRENTARTE A UN BOMBARDEO. TU MISION ES LA DE SUPERAR 26 PRUEBAS DIFERENTES PARA PODER ENTRAR A LA SALA DEL PLACER. PRESENTACION DE LUJO QUE INCLUYE 2 CASSETTES Y EL TEMA N.º 1 EN U.K. "RELAX" EN UNA GRABACION ESPECIAL.

SPECTRUM/COMMODORE



(BALONCESTO)

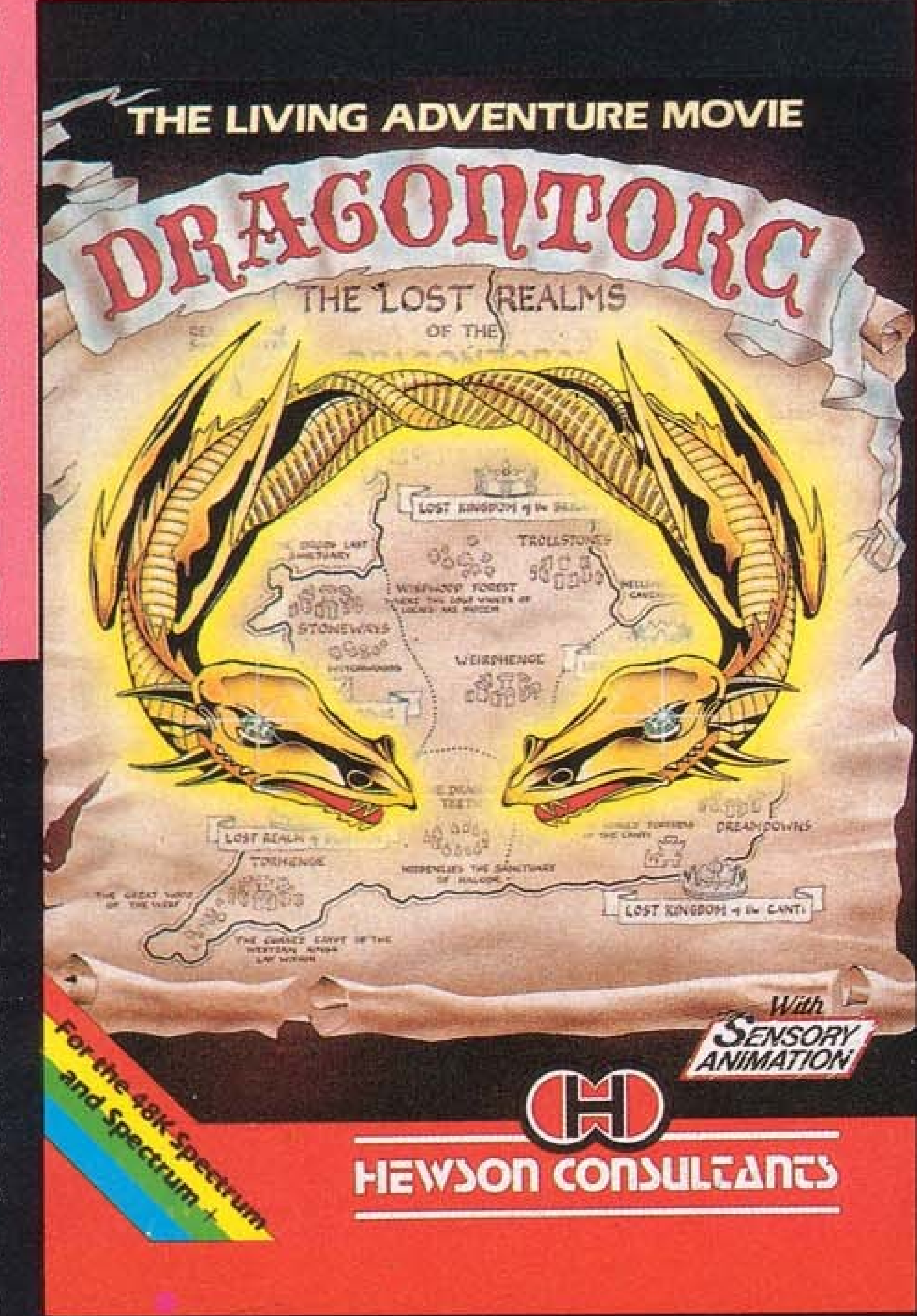
DAS, ESTE ES EL
D DE BALONCESTO
LAS MAQUINAS, EL
COS Y MOVIMIENTO.
ATIS! CON CADA
AGNIFICA CAMISETA
EGALO.
IMITACIONES.

TRUM

DRAGONTROC

UNA PELICULA EN TU SPECTRUM. A TRAVES DE TU VIAJE POR INGLATERRA PODRAS OBTENER PODERES MAGICOS QUE TE AYUDARAN EN TU LUCHA CONTRA LOS MAS DE 80 SINIESTROS PERSONAJES QUE EN CUALQUIERA DE LAS 260 DIFERENTES PANTALLAS SERAN ENVIADOS POR EL MALVADO MORAG CONTRA TI. TODO EL JUEGO ESTA REALIZADO CON UN AUTENTICO EFECTO TRIDIMENSIONAL Y CON UN NUEVO TIPO DE MOVIMIENTO QUE HACE VER A LOS PERSONAJES COMO SI DE UNA PELICULA SE TRATARA.

SPECTRUM/AMSTRAD



ARE, SANTA ENGRACIA, 17, TEL. 447 34 10. 28010 MADRID

correspondiente a la fila del coste material bruto, el coste total quedará adecuadamente ajustado y se producirán los cambios necesarios en el beneficio total. Este es uno de los ejemplos más sencillos, pero algunas hojas de trabajo son capaces de realizar cálculos enormemente complejos. Esto las hace muy útiles para responder a preguntas del tipo «¿qué sucedería si...?», que constantemente necesitan una respuesta en los negocios, y en muchas otras áreas. Aunque la hoja de trabajo se utiliza principalmente para fines comerciales, también puede servir, por ejemplo, para predecir los cambios de población. De hecho cualquier situación en la que hay muchos valores variables independientes, constituye una potencial aplicación para una hoja de trabajo.

La potencia y versatilidad de las hojas de trabajo está haciendo que se convierta en uno de los tipos de *software* que mejor se venden. Muchas hojas de trabajo son compatibles con otro *software,* de forma que se puede constituir una biblioteca completa de *software* para aplicaciones más serias. Si se juntan un procesador de textos, un sistema de gestión de base de datos y una hoja de trabajo, resulta un «paquete» de programas enormemente potente, como por ejemplo el «Simphony», diseñado para el PC de IBM y compatibles.

UNA HOJA DE TRABAJO TIPICA

La unidad básica de la hoja de trabajo es la celdilla. El contenido de cada una puede ser una matriz unidimensional —por ejemplo una palabra— un número, o una fórmula. Cuando la hoja de trabajo se carga en el ordenador, las celdillas tienen una cierta anchura preseleccionada. En algunas hojas de trabajo, este valor atribuido «por defecto» se puede cambiar a voluntad, ya sea al comienzo de la sesión o en un momento posterior.

El valor representado en cada casilla puede ser un número que tú introduces o el resultado de un cálculo. El número de filas y columnas varía de una hoja de trabajo a otra, pero un valor corriente en las hojas de trabajo comerciales serias es 65 columnas y 256 filas. Esto supone 16.640 casillas individuales, lo cual es mucho trabajo para un micro.

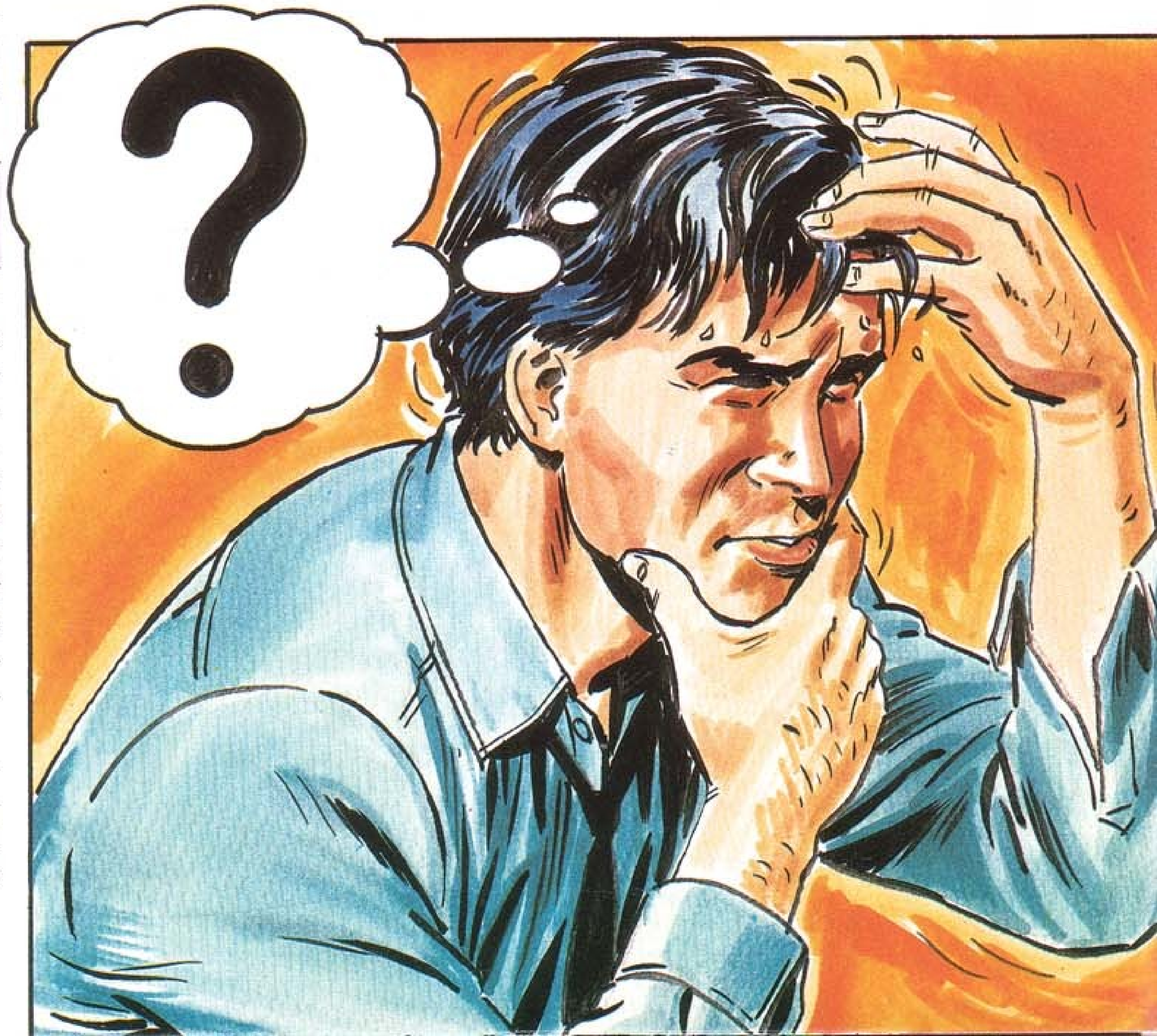
El direccionamiento y localización de casillas se hace siempre por medio de letras o números a lo largo de los ejes X e Y de la cuadrícula, pero los detalles exactos de cómo se hace esto varían de una hoja de trabajo a otra. La mayoría utilizan una combinación de letras y números con las columnas designadas por A, B, C, ..., Z, siguiendo después con AA, AB, AC, ..., AZ, etc., para grandes hojas de trabajo. En tal caso las filas se designarían por números empezando por el 1. Este es el procedimiento seguido en el programa que sigue. La nomenclatura resulta entonces similar a la utilizada en los juegos de barquitos.

Hay varios comandos disponibles para introducir ecuaciones, valores o etiquetas, copiar celdillas o mirar diferentes partes de la hoja. Otros comandos sirven para realizar los cálculos y te permiten cargar y extraer los datos. Las ecuaciones pueden manejar las operaciones aritméticas usuales —sumar, restar, multiplicar y dividir— así como tantos por ciento y totales en cualquier fila y columna.

PLANIFICACION Y DISEÑO

La primera fase de la utilización de una hoja de trabajo es una de las más difíciles. Con frecuencia, requiere mucha planificación, pero durante ella no se utiliza el ordenador. Antes de empezar, tienes que decidir exactamente lo que quieres que haga el ordenador, ya que esto afectará a la manera de diseñar tu hoja de trabajo. Una hoja de trabajo correctamente concebida constituye un método ideal de presentación de información de una manera clara y concisa. Pero, como suele ocurrir con frecuencia en el cálculo, tu hoja de trabjo será lo buena que tú quieras hacerla. Si el trabajo se plantea de una forma descuidada, el resultado será una hoja de trabajo mal presentada, desordenada, difícil de leer, que más bien que revelar la información parece que la esconde.

Como ejemplo práctico, supongamos que quieres confeccionar una hoja de trabajo que te ayude a organizar las finanzas domésticas para todo el año. Evidentemente, tendrás



que utilizar los meses del año como encabezamiento de cada columna, en la parte superior de la hoja. Pero decidir el título de cada fila ya es algo más difícil.

Ante todo, ¿cuál es el grado de detalles que quieres? Algunas partidas obvias de la casa son: hipotecas, contribuciones, calefacción, seguros y mantenimiento. Pero los gastos de reforma y mejora, ¿quieres que se consideren como una categoría independiente? ¿O quieres incluir los costes del coche —gasolina, impuestos, seguro, mantenimiento, reparaciones— en una categoría conjunta, y llamarla algo así como gastos generales? Realmente todo depende de cómo quieras la información de detallada.

Una hoja de trabajo puede ser particularmente útil para el seguimiento del valor de tus inversiones, tales como la casa o el coche. Tendrías que poder averiguar en qué tanto por ciento se está revalorizando tu casa o se está depreciando tu coche.

El cálculo del incremento anual del valor de tu casa parece en principio muy sencillo. El valor correspondiente al año siguiente a la compra será el precio que pagaste por ella multiplicado por el tanto por ciento de crecimiento anual, es decir, P*X%, donde X es 100 más el tanto por ciento anual de incremento. Por ejemplo, para un incremento anual de 0,5%, X sería 100,5%. La fórmula para obtener el valor al cabo del segundo año sería P*X%*X%. La fórmula se hace aún más larga para el tercer año y para el final del décimo año es ya imposible de manejar.

Con una hoja de trabajo hay un procedimiento más sencillo. Afortunadamente no tienes que ser un matemático que domine docenas de complicadas fórmulas, para sacarle todo su jugo a la hoja de trabajo. En un caso como éste puedes utilizar la dirección de una celdilla para referirte al contenido de la misma. En este caso, la complicación de la fórmula no pasa de P*X%, donde P es el contenido de la celdilla anterior. En la fórmula que escribirías para la hoja de trabajo, P sería realmente la dirección de la celdilla anterior y podría ser algo como B10*100,5%.

Si introduces la fórmula en la celdilla C10 la respuesta aparecerá en dicha celdilla. Si introduces en la celdilla D10 la fórmula C10*100,5%, el ordenador toma el número representado en C10 y lo multiplica por 100,5%. La forma real que han de adoptar las ecuaciones varía de una hoja de trabajo a otra, y los programas que siguen utilizan un método algo diferente. Sin embargo los detalles concretos se explicarán en las instrucciones sobre cómo usar el programa, que veremos más adelante.

El uso de la dirección de cada celdilla en vez del contenido de la misma, hace que resulte muy fácil trabajar con una hoja de trabajo. Hace que casi cualquiera pueda efectuar cálculos matemáticos muy complicados con la ayuda de un poco de paciencia y sentido común.

¿QUE SUCEDERIA SI...?

Tu hoja de trabajo para el presupuesto doméstico te permitirá responder a toda clase de preguntas del tipo ¿qué sucedería si...? ¿Qué ocurriría si la hipoteca aumenta en junio un 2%? ¿Y si me compro un coche? ¿Y si instalamos calefacción central y/o dobles ventanas? Y así sucesivamente.

De hecho, este último ejemplo apunta hacia otra área donde se pueden utilizar las hojas de trabajo, aparte de las previsiones financieras y presupuestarias. Con una sola ojeada puedes ilustrar la diferencia entre sistemas de calefacción central que utilicen combustibles distintos. Y si puedes estimar la pérdida de calor que se evitaría con el uso de dobles ventanas, podrías encontrar lo que te ahorrarías y el tiempo que se tardaría en amortizar el coste de la instalación. La hoja de trabajo te puede valer incluso para dimensionar el tamaño correcto y el número de radiadores requeridos para calentar el piso o la casa a la temperatura adecuada.

OTROS POSIBLES USOS

Aunque hasta la más sencilla hoja de trabajo se puede utilizar en aplicaciones muy serias y complicadas, también pueden resultar divertidas. Igualmente, se pueden construir modelos distintos de los financieros usuales.

En el nivel más sencillo, y como puro entretenimiento, se puede crear una referencia circular por todas las casillas, que siga circulando indefinidamente.

Como ocurre con el *software* de utilidades muy difundido, de una hoja de trabajo a otra puede haber enormes variaciones de diseño y complejidad.

Como norma general, se puede decir que cuanto más potente es una hoja de trabajo, más caro resultará y más grande será el microordenador para el que esté diseñado.

TECLEA EL PROGRAMA

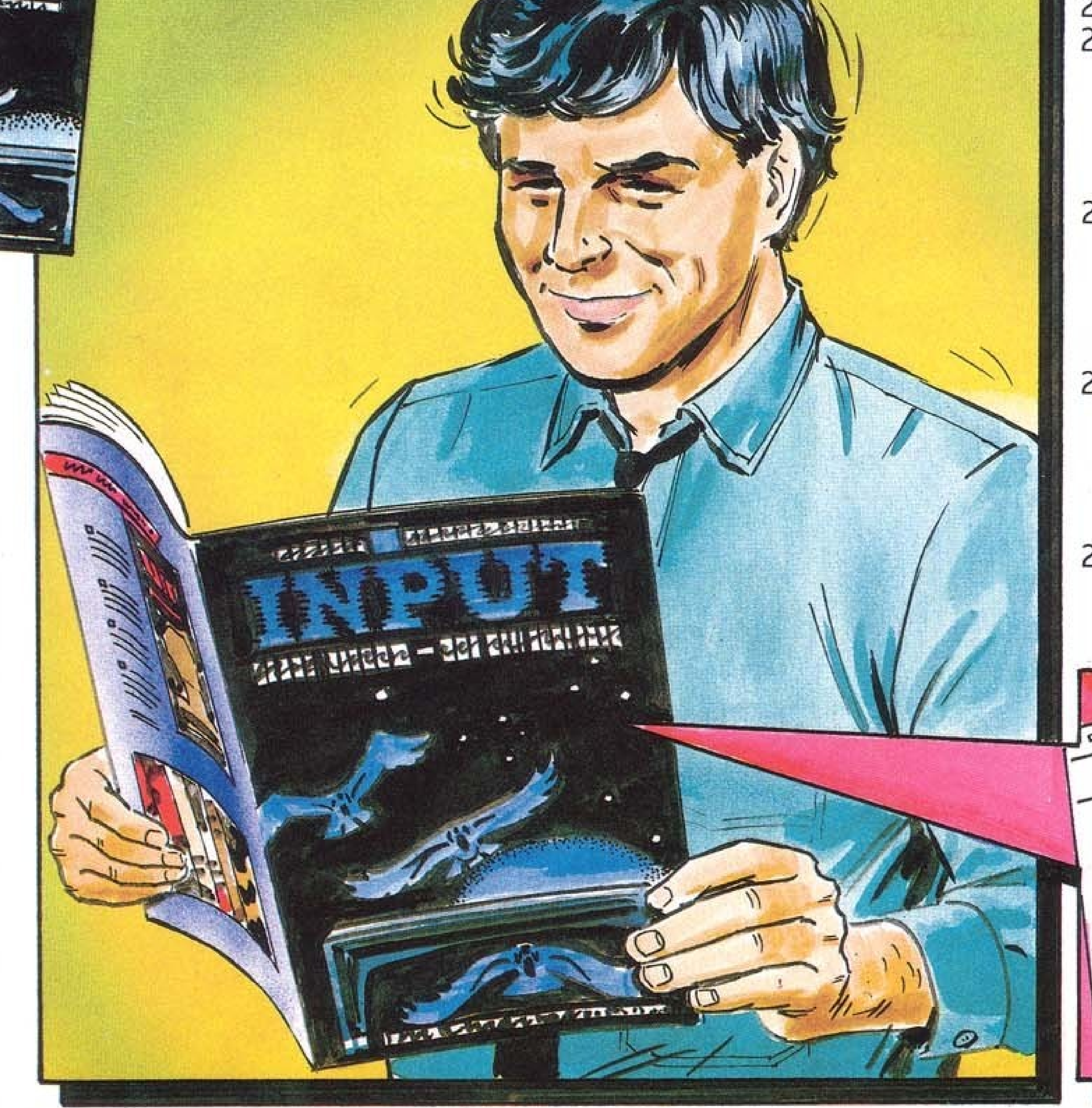
El programa de la hoja de trabajo es muy largo, por lo que lo publicaremos en tres partes. De momento, puedes ir copiando las líneas que damos a continuación y almacenándolas de forma que se puedan añadir más líneas posteriormente.

Teclea

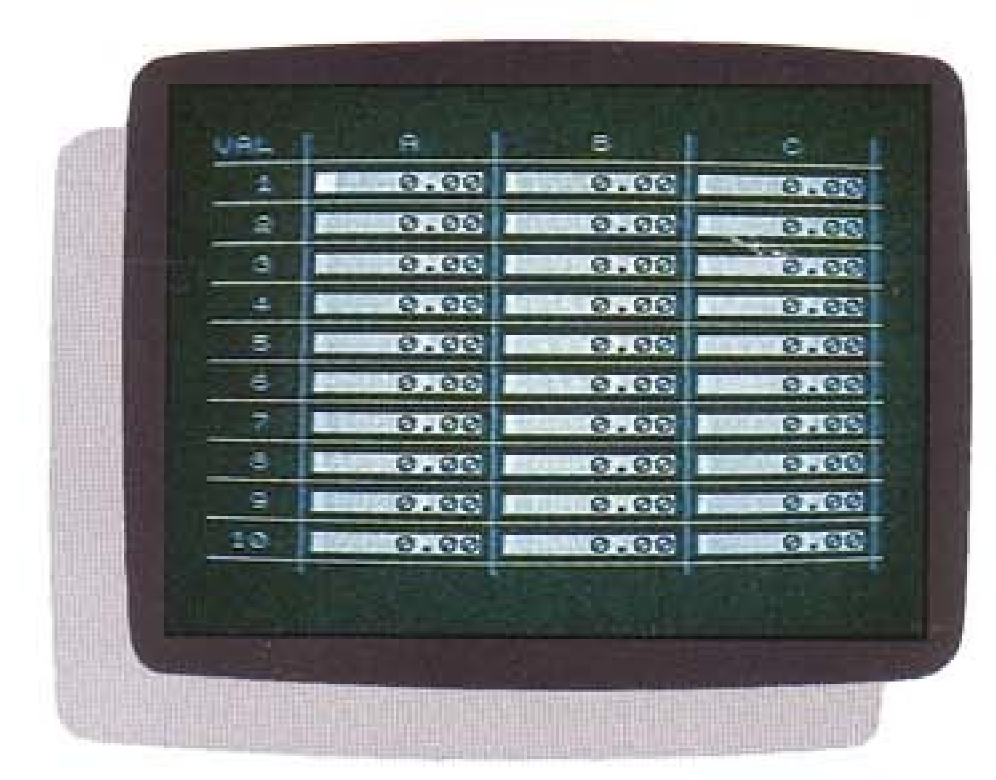
```
5 BORDER Ø: PAPER Ø: INK 7:
  CLS
10 DIM b$(11): DIM s$(8):
   DIM d$(3Ø,24,18): DIM
   v(4): DIM z$(5,4)
2Ø GOSUB 173Ø: POKE
   23658,8: LET t$="VAL":
   LET os=Ø: LET sflag=Ø:
   LET wx=1: LET wy=1: LET
   cx=1: LET cy=1
3Ø CLS: PRINT"□□□□□□
   □□□□□□□□□□□□□□□□
```



```
□□□□□□□": FOR x=4 TO 32
   STEP 9: FOR y=2 TO 21
   STEP 2: PRINT AT y,x;"":
   NEXT y: NEXT x: FOR y=1
   TO 21 STEP 2: PRINT AT
   y,0;"": NEXT y
40 FOR X=0 TO 2: PRINT AT
   0,9*x+9;CHR$ (wx+x+64):
   NEXT x
50 FOR x=0 TO 9: PRINT AT
   (x+1)*2,1;("□" AND
   wy+x<10);wy+x: NEXT x
60 PRINT AT 0,0;t$;"□":
   FOR y=0 TO 9: FOR x=0 TO
   2: GOSUB 1230: NEXT x:
   NEXT y: PRINT ‘1; AT 0,0;"
   □□□□□□□□□□□□□□□□□
   □□□□□□□□□□□□□□□□"
70 PRINT AT
   cy*2,((cx-1)*9)+5;
   BRIGHT 1; FLASH 8; PAPER
   8; INK 8; OVER 1;"□"
80 IF INKEY$="'" AND wy>1
   THEN LET wy=wy-10: GOTO
   40
90 IF INKEY$="&" AND wy<20
   THEN LET wy=wy+10: GOTO
   40
100 IF INKEY$="(" AND wx<21
   THEN LET wx=wx+3: GOTO
   40
110 IF INKEY$="%" AND wx>1
   THEN LET wx=wx-3: GOTO
   40
120 PRINT AT
   cy*2,((cx-1)*9)+5;
   FLASH 8; BRIGHT 0; INK 8;
   PAPER 8; OVER 1;"□"
130 LET cy=cy+(INKEY$="6"
   AND cy<10)-(INKEY$="7"
   AND cy>1): LET
   cx=cx+(INKEY$="8" AND
   cx<3)-(INKEY$="5" AND
   cx>1)
140 IF INKEY$="i" OR
   INKEY$="I" THEN GOSUB
   1250
150 IF INKEY$="v" OR
   INKEY$="V" THEN LET
   t$="VAL": GOTO 60
160 IF INKEY$="e" OR
   INKEY$="E" THEN LET
   t$="EQU": GOTO 60
170 IF INKEY$="?" THEN PRINT
   AT 0,0; FLASH 1;"CALC":
   GOSUB 810: GOTO 60
180 IF INKEY$="z" OR
   INKEY$="Z" THEN PRINT AT
   0,0; FLASH 1;"COPY":
   GOSUB 230: GOTO 60
190 IF INKEY$="P" OR
   INKEY$="p" THEN COPY
200 IF INKEY$="NOT" THEN
   GOSUB 1780: GOTO 30
210 IF INKEY$="-" THEN GOSUB
   1840: GOTO 30
220 GOTO 70
230 PRINT ‘1;AT 0,0;"CELDILLA
   A COPIAR?": LET d=1: LET
   c=3: LET x=15: GOSUB
   580: GOSUB 670: IF f THEN
   BEEP .2,30: GOTO 230
240 PRINT ‘1;AT 0,0;"ABS O
   REL (A O R)?: LET x=22:
   LET d=2: LET c=1: GOSUB
   580: GOSUB 670: IF f THEN
   BEEP .2,30: GOTO 240
250 PRINT ‘1:AT 0,0;"COL O
   FIL (C O R)": LET x=22:
   LET d=3: LET c=1: GOSUB
   580: GOSUB 670: IF f THEN
   BEEP .2,30: GOTO 250
260 PRINT ‘1; AT 0,0;"DESDE
   CELDILLA NO?": LET x=16:
   LET d=4: LET c=3: GOSUB
```



```
    580: GOSUB 670: IF f THEN
    BEEP .2,30:
    GOTO 260
270 PRINT #1;AT 0,0;"HASTA
    CELDILLA NO?": LET x=14,
    LET d=5: LET c=3: GOSUB
    580: GOSUB 670: IF f THEN
    BEEP .2,30: GOTO 270
280 GOSUB 770: IF NOT f THEN
    GOTO 320
290 PRINT #1;AT 0,0;"ERROR
    COMANDO: PRESIONA A PARA
    CANCELAR O CUALQUIER
    TECLA PARA REINTRODUCIR"
300 PAUSE 10: PAUSE 0: PRINT
    #1;AT 0,0;"□□□□□□□□□□
    □□□□□□□□□□□□□□□□□□
    □□□□□□□□□□□□□□□□□□
    □□□□□□□□□□□□□□□□□□
    □□□□□□□□□□□□□□□□□□
    □□□": IF INKEY$="a"OR
    INKEY$="A"THEN RETURN
310 GOTO 230
320 LET a=(CODE z$(1,2))-64:
    LET b=VAL z$(1,3 TO (VAL
    z$(1,1)+1)): LET
    s$=(d$(b,a,9 TO 16) AND
    t$="ECU")+(d$(b,a, TO 8)
    AND t$="VAL"): LET
    c$=d$(b,a,17): LET
    z=CODE d$(b,a,18)
330 IF z$(2,2)="R" AND
    T$="ECU" AND C$="1" THEN
    GOTO 390
340 FOR a=fc TO tc: FOR b=fr
    TO tr
350 IF t$="ECU" THEN LET
    d$(b,a,9 TO 16)=s$: LET
    d$(b,a,17)=c$: LET
    d$(b,a,18)=CHR$ Z
360 IF t$="VAL" THEN LET
    d$(b,a, TO 8)=s$
370 NEXT b: NEXT a
380 RETURN
390 LET s$=d$(b,a,9 TO 16):
    GOSUB 890: LET a=(CODE
    z$(4,2)-64)-(CODE
    z$(1,2)-64): LET b=VAL
    z$(4,3 TO (VAL
    z$(4,1)+1))-VAL z$(1,3
    TO (VAL z$(1,1)+1))
400 LET v(2)=v(2)+b-1: LET
    v(4)=v(4)+((b-1) AND
    v(3)<>26)
410 LET v(3)=v(3)+((a-1) AND
    v(3)<>26): LET
    v(1)=v(1)+a-1
412 IF z$(3,2)="C" THEN LET
    v(1)=v(1)+1: LET
    v(3)=v(3)+(v(3)<>26)
414 IF z$(3,2)="R" THEN LET
    v(2)=v(2)+1: LET
    v(4)=v(4)+(v(4)<>26)
```



Hoja de trabajo con su característica división en casillas.

RUTINAS DE TANTEO Y TIEMPO

No hay nada como saber que sólo dispones de dos segundos para regresar a tu base, o que te faltan diez puntos para obtener la máxima puntuación en tu juego de batallas. Todos los juegos de marcianitos llevan algún tipo de cuenta de tanteo y tiempo, para darle más emoción. Con algunos programas sencillos, tú puedes hacer lo mismo.

Casi todos los juegos de ordenador necesitan algún tipo de puntuación o temporización, o incluso ambas cosas. Sin ellos no puedes juzgar lo bien que se te da el juego, o si vas mejorando algo, y no suele tener mucho interés jugarlo con tus amigos.

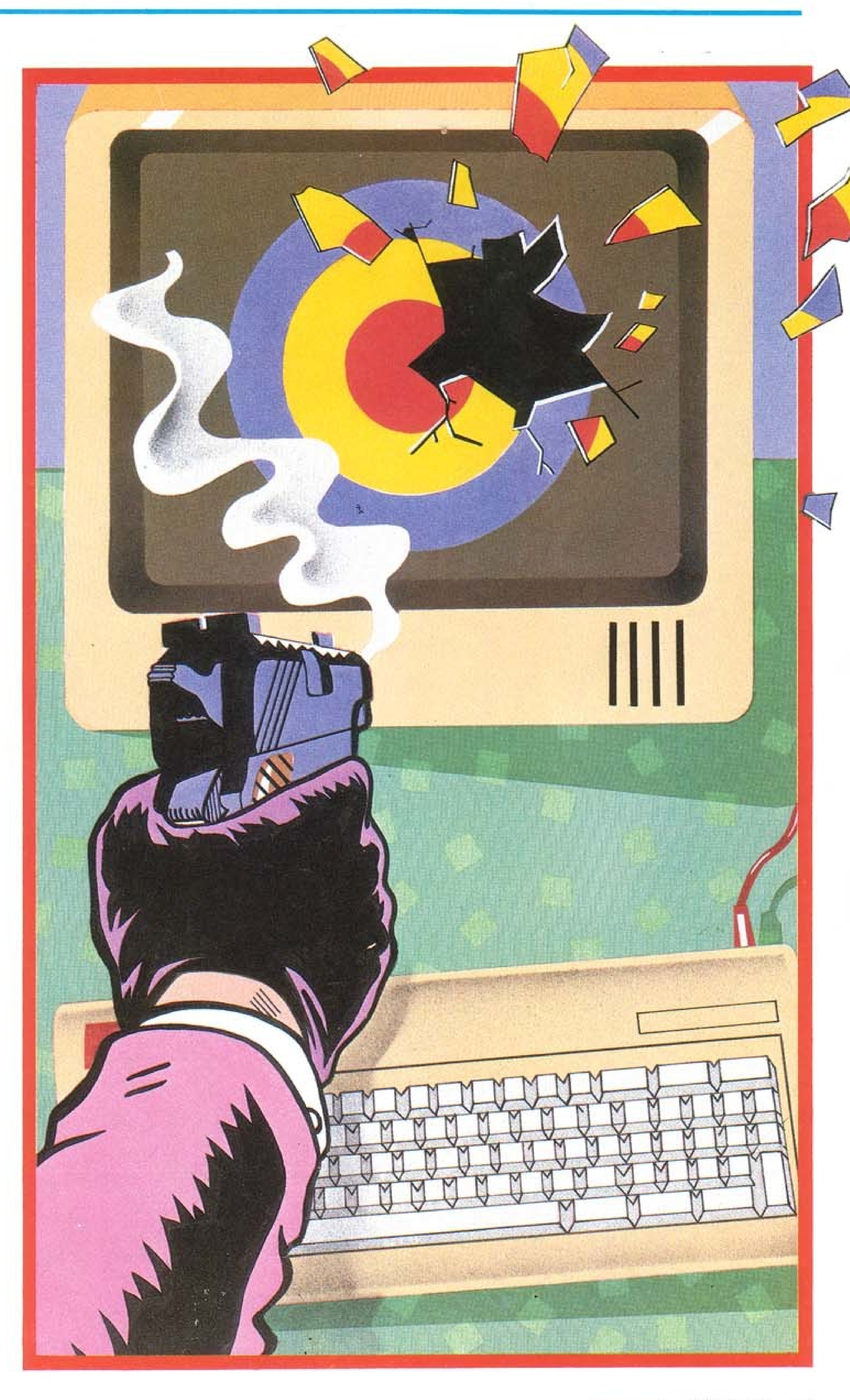
Podrías tener a alguien sentado tras de ti que fuera contando los impactos que consigues sobre tu feroz enemigo, pero esto no tiene mucho sentido cuando puedes programar a tu ordenador para que los cuente él. Con unas cuantas líneas de programa más, la máquina recordará también las puntuaciones.

Por la misma razón no hay necesidad de recurrir a un cronógrafo para la medida del tiempo. Todas las máquinas llevan un reloj incorporado, y puedes servirte de él de muchas formas para mejorar tus juegos.

CAMPO DE MINAS

Para que veas la manera de incorporar en la práctica las rutinas de puntuación y temporización, aquí tienes un juego en el que se van añadiendo las rutinas progresivamente. Cada rutina es muy sencilla y se puede añadir también a otros juegos.

El juego se llama Campo de Minas, y en él tú vas conduciendo un carro de combate, cuya misión es rescatar a unos paracaidistas que se han arrojado temerariamente sobre un campo minado. Cada vez que el tanque se



mueve, corre el riesgo de hacer detonar una mina plantada aleatoriamente por el ordenador. Como en un campo minado de verdad, las minas son invisibles, por lo que tendrás que moverte con precaución.

El tanque (desafortunadamente es sólo un signo #, hasta que aprendas la manera de combinar gráficos y movimiento en un programa BASIC) se controla utilizando las teclas normales de movimiento: Z para izquierda, X para derecha, P hacia arriba y L abajo. De hecho, el núcleo del programa está constituido por la rutina de "moverse por la pantalla", que ya conoces.

Cuando teclees esta sección del juego y la ejecutes (con RUN), verás que todavía no está completa: después de que hayas rescatado al paracaidista, no sucede nada, excepto que el tanque continúa vagando sin rumbo por la pantalla. El programa ha de detenerse pulsando la tecla BREAK, o tendrás que esperar hasta que el tanque tropiece con una mina escondida. Pero no te alarmes, todo esto mejorará en cuanto le pongas las rutinas de puntuación y de tiempo que siguen.

```
 50 LET tx=16: LET ty=5
 60 CLS
 70 PRINT AT 11,0;"------
    --------------------
    -------"
 90 LET px=INT (RND*30)+1
100 LET py=INT (RND*10)
110 IF px=tx AND py=ty THEN
    GOTO 90
120 PRINT AT py,px;"O";AT
    ty,tx;"#"
130 LET txx=tx: LET tyy=ty
140 LET a$=INKEY$
145 IF a$="p" THEN LET
    ty=ty-1
150 IF a$="l" THEN LET
    ty=ty+1
160 IF a$="z" THEN LET
    tx=tx-1
170 IF a$="x" THEN LET
    tx=tx+1
190 IF ty<0 OR ty>10 THEN LET
    ty=tyy
200 IF tx<1 OR tx>30 THEN LET
    tx=txx
230 PRINT AT tyy,txx;"□"
240 PRINT AT ty,tx;"#"
250 LET mx=INT (RND*30)+1
260 LET my=INT (RND*10)
270 IF mx=tx AND my=ty THEN
    PRINT AT my,mx;"□":
    PRINT AT 8,3; "BOOM!!-
    TE ALCANZO UNA MINA": STOP
310 GOTO 130
```

En este programa, la pantalla queda dividida en dos mitades por la línea 70, que imprime una serie de rayas en la misma. Las líneas 145 a 170 controlan el movimiento del tanque por la pantalla. En el capítulo "Movimiento y animación" tienes una explicación completa de cómo se hace esto.

Las líneas 190 y 200 mantienen el tanque confinado en una zona por encima de la línea de trazos (dibujada por la línea 70), evitando así que se salga de la pantalla.

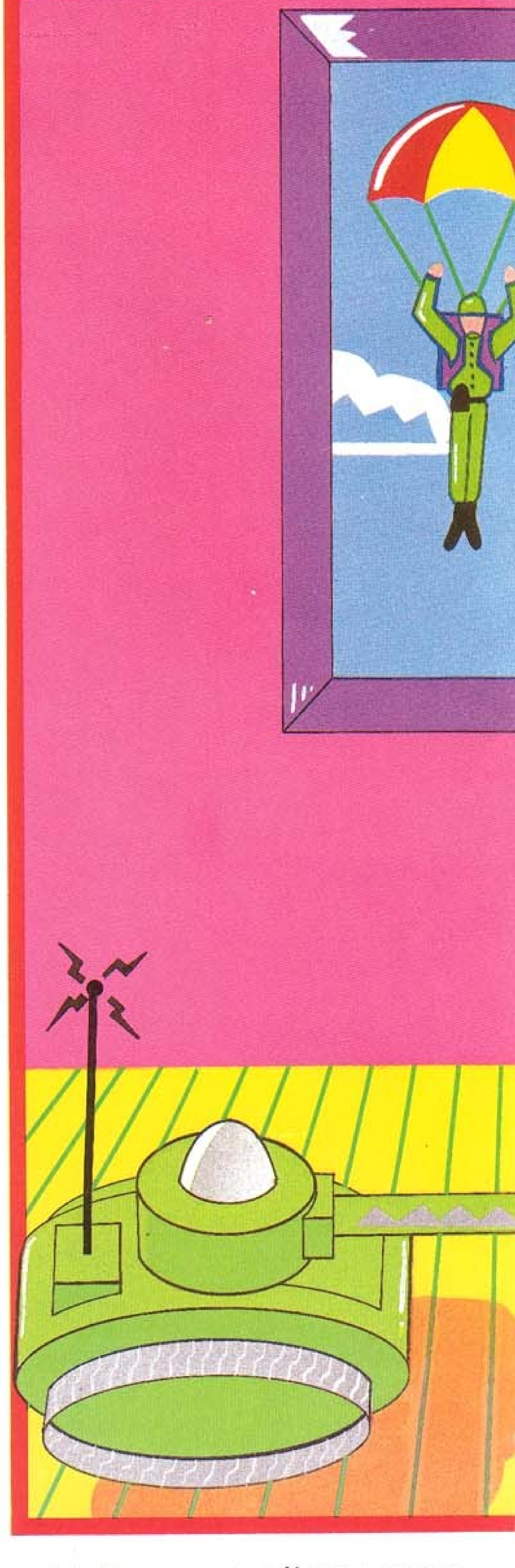
La posición de caída de los paracaidistas se elige aleatoriamente en las líneas 90 y 100. En la línea 110 se comprueba si el tanque y el paracaidista ocupan la misma posición en la pantalla. Si así es, se elige una nueva posición de caída en la línea 90.

De una forma análoga, en las líneas 250 y 260 se elige un emplazamiento para la mina. La línea 270 compara la posición de la mina con la del tanque. Si ambas coinciden, el tanque es eliminado de la pantalla y se representa un mensaje de explosión. El programa se detiene entonces.

PUNTUACION

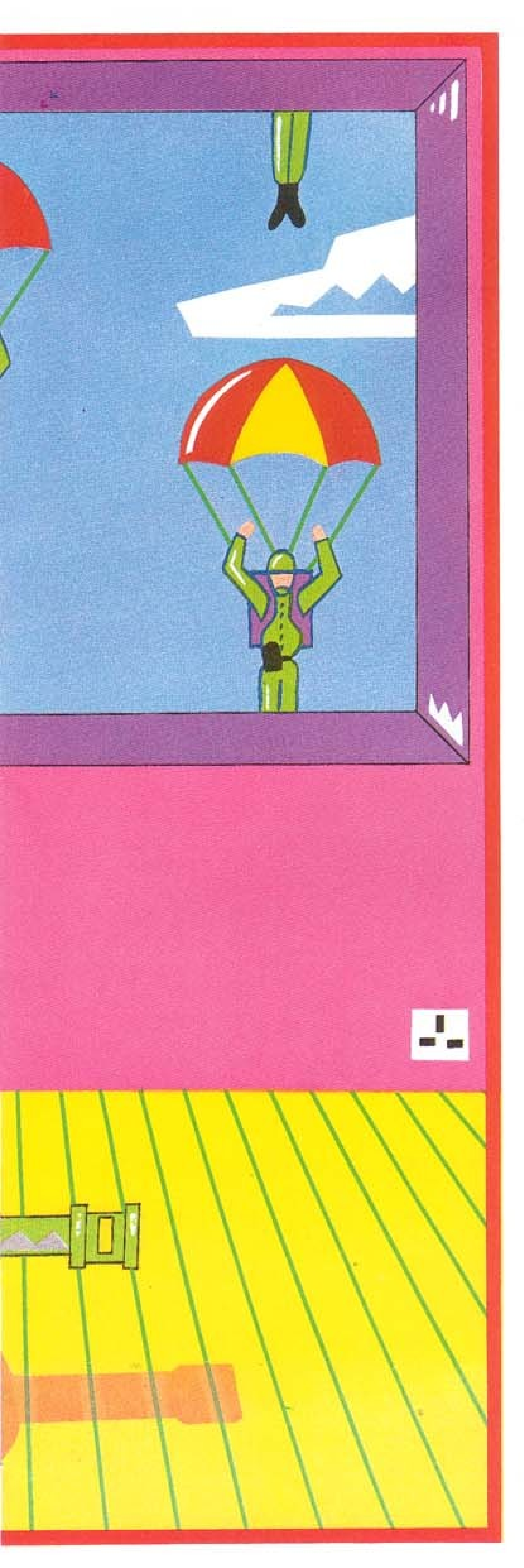
En los juegos de extraterrestres, la puntuación aumenta normalmente cuando la posición ocupada por dos objetos en la pantalla es la misma. Los objetos pueden ser un misil y un blanco, un comecocos y una píldora alimenticia, un tanque y un paracaidista, un caballo y una meta, o lo que el juego requiera.

```
 40 LET s=0
280 IF px=tx AND py=ty THEN
    LET s=s+1:GOTO 90
330 PRINT AT 14,8;s;
    "□PARACAIDISTAS"
```



Añade, pues, estas líneas a tu programa y podrás comprobar cómo trabaja en la práctica el mecanismo de puntuación.

Cambia el STOP de la línea 270 por un GOTO 330. La línea 270 se convierte ahora en:



```
270 IF mx=tx AND my=ty THEN
    PRINT AT my,mx;"!": PRINT
    AT 8,3;"BOOM!!-TE ALCANZO
    UNA MINA": GOTO 330
```

La línea 280 es importante. En ella se comprueba que el tanque y el paracaidista ocupan la misma posición en la pantalla. Si ocurre esto, la puntuación aumenta en 1.

En la línea 40 se pone a cero el tanteo antes de que empiece el juego y la línea 330 visualiza la puntuación. Cambiando el STOP de la línea 270 se consigue que el ordenador presente la puntuación después de haber tocado una mina.

El jugador se enfrenta ahora con una sucesión de paracaidistas a los que rescatar. Cada vez que es rescatado uno, cae otro del cielo. El juego se detiene cuando se produce la explosión de una mina (porque el tanque ocupa la misma posición en la pantalla).

PUNTUACION MAXIMA

No es difícil añadirle a tu juego una opción de puntuación máxima. No tienes más que introducir una variable asociada a esta puntuación —por ejemplo, HS— y algún método para actualizarla cuando sea superada dicha opción, además hay que poner una rutina de presentación.

Aquí tienes las líneas que debes añadir para tener una opción de puntuación máxima.

```
 30 LET hs=0
350 IF s>hs THEN LET hs=s
370 PRINT AT 18,8;"MAX.
    PTOS.=";hs
```

En primer lugar debes poner la puntuación máxima en su menor valor posible, por lo que la línea 30 pone HS a cero. Después de que el juego se ha detenido, la línea 350 compara la última puntuación (S) con la puntuación máxima (HS). Si la anterior puntuación obtenida es mayor que la puntuación máxima se actualiza HS, haciéndola igual a S. Finalmente, la línea 370 visualiza en la pantalla el valor.

Es probable que estas líneas te parezcan suficientes para dotar al juego de un tanteo. Por desgracia, esto no es cierto. Cada vez que le hagas ejecutar el programa (con RUN), el ordenador olvida automáticamente el valor de HS, y los valores de las otras variables. Para mantener el valor de HS tienes que añadir las líneas de "¿Otra vez?" que se describieron en un capítulo anterior.

```
390 PAUSE 100
400 PRINT AT 8,3; TAB 31
410 PRINT AT 8,7;"OTRA VEZ?
    (S/N)"
420 LET a$=INKEY$
430 IF a$="s" THEN GO TO 40
440 IF a$="n" THEN STOP
450 GOTO 420
```

He aquí lo que hacen estas nuevas líneas:

Hay un corto retardo introducido por la sentencia PAUSE. La línea 400 borra una parte de la pantalla dejándola preparada para el mensaje «Otra vez» (S/N) que se presenta en la línea 410.

La rutina "Otra vez" está en las líneas 410 a 450. La línea 450 hace que el programa recomience en la línea 40 si se pulsa S, y la línea 440 detiene el programa si se pulsa N. La línea 450 sirve para asegurarse de que cualquier otra tecla será ignorada.

Como no hay necesidad de pulsar RUN cada vez que desees jugar de nuevo, el valor de HS será preservado, aunque al volver a cargar el programa (con LOAD) se perderá el valor de HS, incluso si encuentras alguna forma de arrancar el programa sin pulsar RUN.

MEDIDA DE TIEMPOS

Tal como está el juego, depende demasiado de la suerte, simplemente, el jugador sigue adelante hasta que pisa una mina escondida.

Se puede introducir en este tipo de juegos un elemento de habilidad, para convertirlo en una carrera contra el reloj. Con los siguientes añadidos puedes cronometrar cuánto tardas en rescatar a diez paracaidistas.

```
 10 LET T=0
 80 POKE 23672,0: POKE
    23673,0
290 IF s<10 AND px=tx AND
    py=ty THEN GOTO 90
```



```
300 IF s=10 THEN GOTO 320
320 LET t=PEEK
    23672+256*PEEK 23673
340 IF s=10 THEN PRINT AT
    16,8; "EN□";t/50;
    "□SEGUNDOS"
```

El reloj interno de la máquina está corriendo todo el tiempo que el ordenador permanece encendido. Tú puedes aprovecharte de esto y para arrancar un contador de tiempos solamente tienes que poner a cero la lectura de dicho reloj. Esto se hace en la línea 80. El objeto de los dos POKEs es poner ceros en dos posiciones de memoria en particular. Lo que ocurre es que la primera posición de memoria cuenta hasta 255 antes de desbordarse hacia la segunda.

El reloj se "detiene" en la línea 320. Realmente, el reloj no puede detenerse; lo que tú haces es que la máquina recuerde una lectura particular en un instante determinado, por ejemplo cuando coinciden dos objetos en la pantalla. A la lectura del reloj se le llama T, por lo que se introduce LET t = PEEK 23672 + 256 * PEEK 23673. Aunque pueda parecerte extraño, lo que hace esta sentencia es mirar directamente los valores de dos direcciones de memoria que se pusieron a cero anteriormente en el programa. La dirección 23672, como cualquier otra dirección en un microprocesador de 8 bits, puede contener números enteros entre 0 y 255. El contenido de la dirección 23673 aumenta 1 cada vez que se agota la dirección 23672.

Así para obtener el tiempo total que el contador del **Spectrum** ha estado contando, tienes que multiplicar el valor de la dirección 23673 por 256 y sumarle el valor almacenado en la dirección 23672.

Ahora puedes ver por qué t es igual a PEEK 23672 + 256 * PEEK 23673. (Ya nos ocuparemos de todos modos ampliamente de PEEK y POKE en un próximo artículo.)

El reloj debe detenerse cuando el jugador ha rescatado a diez paracai-

distas, por lo que en la línea 300 se comprueba si ya se han rescatado diez, en cuyo caso, el programa salta a la línea 320 que es la que "detiene" al reloj.

Si se ha rescatado con éxito a un paracaidistas y, además, el número total de los que ya han sido rescatados es menor que diez, la línea 290 hace caer otro paracaidista.

La línea 340 presenta el tiempo transcurrido para el rescate sólo si se han rescatado ya los diez. La lectura de tiempo es dividida por 50, por lo que el tiempo aparece en segundos. El reloj es actualizado 50 veces cada segundo.

MEJOR TIEMPO

De la misma forma que añadiste antes una opción de tanteo máximo para el juego, puede resultarte interesante una opción de mejor tiempo. En esta variante del juego tienes que poder registrar el tiempo más rápido en que los diez paracaidistas son rescatados, aunque este principio se puede aplicar a cualquier temporizador.

He aquí las líneas que tienes que añadir:

```
 20 LET lt=999999
360 IF t<lt AND s=10 THEN LET
    lt=t
380 PRINT AT 21,4;"TIEMPO
    MINIMO=";LT/50;
    " SEGUNDOS"
```

Igual que con el tanteo máximo se ponía inicialmente un "tanteo máximo" muy bajo, ahora se pone un "tiempo récord" ridículamente largo. La línea 20 asigna a la variable de mejor tiempo (lt) un valor de 999999.

La línea 360 compara el último tiempo obtenido con el mejor tiempo. Si el último tiempo obtenido es más corto que el tiempo récord y además se han rescatado ya diez paracaidistas, entonces se modifica el tiempo récord haciéndolo igual al último tiempo obtenido.

Finalmente, la línea 380 sirve para expresar en segundos el tiempo récord. La variable de tiempo récord es dividida por 50 para que resulte en segundos.

Una cosa que has de recordar es que si estás utilizando una opción de tiempo récord en un juego, tienes que utilizar la rutina de "¿Otra vez?", pues, de lo contrario, el valor del tiempo récord se perderá cada vez que ejecutes (con RUN) el programa.

EL TECLADO Y EL COMPUTO DEL TIEMPO

Hasta ahora has visto cómo puedes controlar el reloj interno de la máquina desde dentro de un programa, examinando las posiciones de dos objetos sobre la pantalla. Otra forma de "detener" el reloj es servirte del teclado.

Lo puedes hacer utilizando la sentencia INKEY$. Resulta tan fácil como arrancar y parar un cronómetro para controlar el movimiento de los objetos por la pantalla.

Aquí tienes un juego de acción rápida que ilustra cómo puede usarse el teclado para detener el reloj:

```
 20 CLS
 30 LET n=INT (RND*400)+1
 40 PAUSE n
 60 PRINT AT 14; "DISPARA!!"
 70 POKE 23672,0: POKE
    23673,0
 80 IF INKEY$=""THEN GOTO 80
 90 LET T=PEEK
    23672+256*PEEK 23673
100 PRINT AT 11,14;"BANG!!"
110 PAUSE 50
130 LET m=INT (RND*35)+1
140 IF t<m THEN PRINT AT
    11,9; "HAS SOBREVIVIDO"
150 IF t>m THEN PRINT AT
    11,11; "ESTAS MUERTO"
```

El programa presenta el mensaje "TIRA!!" y el jugador ha de pulsar cualquier tecla tan rápido como pueda. Se mide el tiempo de reacción desde el momento en que apareció el mensaje y el ordenador informa al jugador de quién ha ganado, si el ordenador o él.

El programa es muy sencillo. Después de que la línea 20 ha borrado la pantalla, las líneas 30 a 50 introducen una pausa aleatoria. La línea 60 sirve para enviar el mensaje "TIRA!!" e inmediatamente se arranca el contador de tiempo en la línea 70. La línea 80 hace que la máquina espere, continuando cuando se ha pulsado una tecla cualquiera. Ya vimos esta línea al ocuparnos del "Control del Teclado".

En cuanto se ha pulsado una tecla cualquiera, el programa continúa a la línea 90, que realmente para el contador, llamando T a la lectura que tiene en ese momento. La línea 100 presenta el mensaje "BANG!!". Hay una pausa introducida por la línea 110 antes de que la máquina elija un instante de tiro. La línea 130 es la que hace esto.

La máquina tiene ahora dos variables, tu tiempo, T, y el tiempo de la máquina, M. Las líneas 140 a 160 comparan estos valores y presentan el resultado del duelo.

P y R

¿Existe algún límite para la duración máxima que se puede tener?

Sí, existe un límite, aunque normalmente es tan alto que en la práctica no tiene importancia. El reloj interno de casi todos los ordenadores domésticos avanza a la misma velocidad, y el factor limitador es la cantidad de pulsos de tiempo que el ordenador puede recordar. En el **Spectrum** se pueden tener hasta tres bytes, lo que equivale a unos cuatro días.

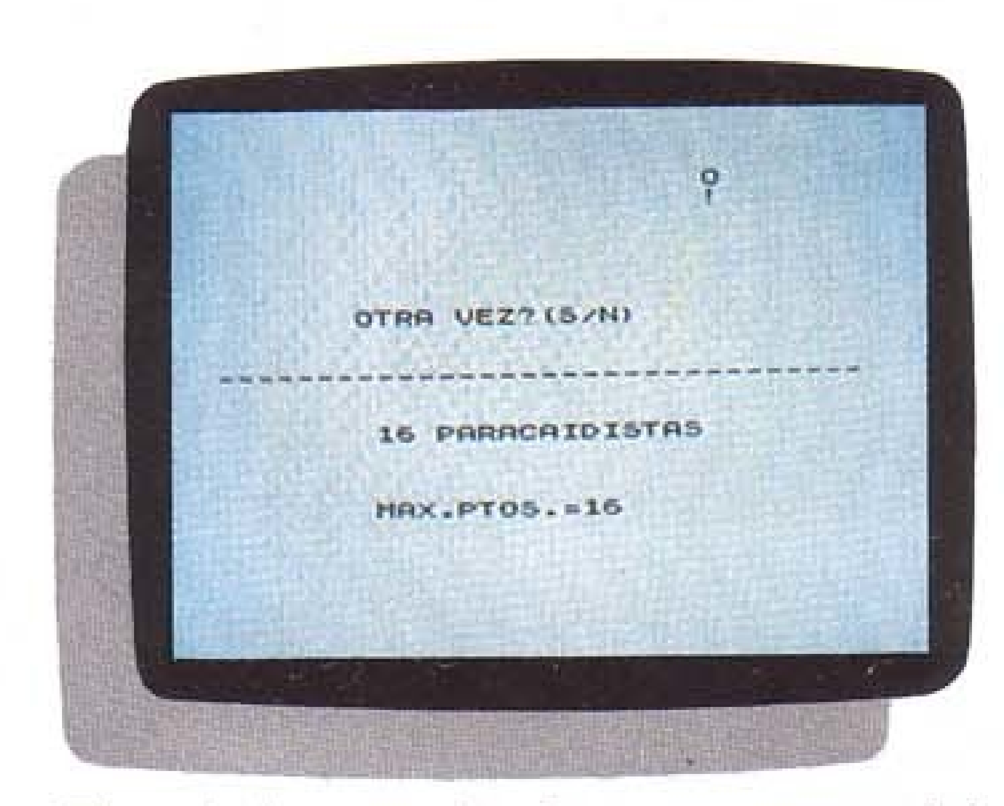


Foto de la pantalla de un momento del juego.

JUEGOS DE LABERINTO

- LOS PRINCIPIOS DE LA ANIMACION
- MOVIMIENTO DE GRAFICOS
- COMO UTILIZAR LOS GRAFICOS INCORPORADOS

Los juegos de laberintos sofisticados requieren programas largos. Pero tú puedes diseñar algunos sencillos extrayendo de ellos importantes principios y utilizando poco más que un bucle y sentencias DATA.

Los juegos de laberintos ejercen una fascinación permanente sobre los propietarios de un ordenador, por lo que las casas de *software* continúan sacando nuevas variantes del comecocos.

Este artículo te enseñará la manera de saltar al carro de los fabricantes de laberintos, permitiendo que te construyas el tuyo propio.

En su primera fase el laberinto no incluye «enemigos» ni obstáculos, ya que esto requeriría un programa muy largo. Pero te enseñará cómo se programa el que tu carácter principal no pueda atravesar las paredes, lo cual es la base de todos los juegos de esta clase. También se incluye la puntuación y el crono, así como una rutina de «mejor tanteo», a fin de darle un cierto interés competitivo.

La manera más fácil de entender cómo funciona el juego del laberinto es ir introduciéndolo por etapas. Así, pues, empieza construyendo el propio laberinto:

```
100 FOR n=3 TO 17
110 READ a$
120 FOR m=7 TO 21
130 PRINT AT n,m;"."
140 IF a$(m-6)="p" THEN
    PRINT PAPER 1; INK 1;AT
    n,m;"—"
150 NEXT m
160 NEXT n
9000 DATA "ppppppppppppppp"
9010 DATA "p.............p"
9020 DATA "p.pp.pp.pp.pp.p"
9030 DATA "p.p.........p.p"
9040 DATA "p...p.p.p.p...p"
9050 DATA "p.ppp.p.p.ppp.p"
9060 DATA "p.....p.p.....p"
9070 DATA "pppp.pp.pp.pppp"
9080 DATA "p.....p.p.....p"
9090 DATA "p.ppp.p.p.ppp.p"
9100 DATA "p...p.p.p.p...p"
9110 DATA "p.p.........p.p"
9120 DATA "p.pp.pp.pp.pp.p"
9130 DATA "p.............p"
9140 DATA "ppppppppppppppp"
```

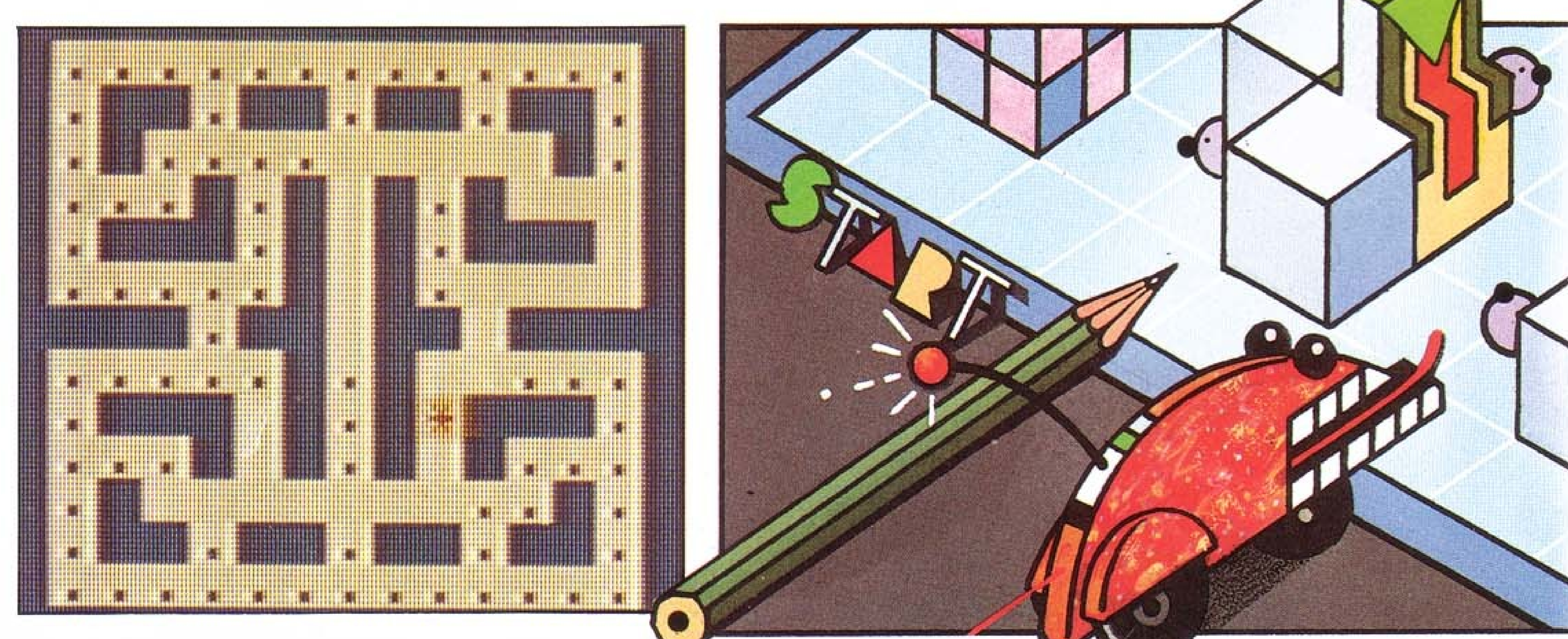
Las líneas 100, 120, 150 y 160, que definen un par de bucles FOR...NEXT, establecen los contornos del laberinto. Con la línea 130 se imprime un punto en cada cuadrado.

Las líneas 110 y 140 se encargan de leer los datos de las líneas 9000 a 9140 y de sustituir el punto por un espacio, pero con tinta azul sobre el papel azul cada vez que un elemento en un DATA presente la letra p. Pueden hacerlo así porque normalmente el **Spectrum** no presentará dos caracteres en la misma posición, de forma que el segundo sustituirá al primero.

CONSTRUYENDO EL «COMILON»

Sin embargo, un laberinto resulta bastante inútil si no hay algo que se mueva a través de él. Ejecuta pues (con RUN) el programa, y anádele lo siguiente:

```
50 LET x=10
60 LET y=14
```



```
1000 PRINT PAPER 6; INK 2;AT
     x,y;"*"
1010 LET xx=x
1020 LET yy=y
1030 IF INKEY$=""THEN GOTO
     1030
1040 IF INKEY$="p" AND ATTR
     (x-1,y) <>9 THEN LET
     x=x-1
1050 IF INKEY$="i" AND ATTR
     (x+1,y) <>9 THEN LET
     x=x+1
1060 IF INKEY$="z" AND ATTR
     (x,y-1) <>9 THEN LET
     y=y-1
1070 IF INKEY$="x" AND ATTR
     (x,y+1) <>9 THEN LET
     y=y+1
1080 PRINT INK 7;AT xx,yy;" "
1090 GOTO 1000
```

Si ya has trabajado con el control por teclado, la mayor parte de estas líneas te resultarán familiares. Como en los famosos juegos de caracteres, nuestro amigo el asterisco es colocado en x, y moviéndose después —con tu ayuda— merced a una serie de líneas con INKEY$.

No obstante aquí hay una diferencia esencial: Se puede trasladar a un determinado cuadrado solamente si éste *no* es azul. O lo que es lo mismo, si ATTR *no* es igual a 9. Aquí 9 es el valor numérico correspondiente a los colores en que astutamente has dibujado las líneas de tu laberinto.

Si quieres crear un laberinto de otro color, aquí tienes la forma de obtener el valor adecuado de ATTR:

1. Toma el código de color de INK, tal como aparece en el teclado del ordenador, en este caso es 1. Resultado, 1.
2. Toma el código del color de PAPER, en este caso 1, y multiplícalo por 8. Resultado, 8.
3. Súmale 64 si el área de interés BRIGHT. En este caso el resultado es 0.
4. Suma 128 si el área de interés es FLASH. En este caso el resultado es 0.

A continuación suma todos los números anteriores. El resultado es el número ATTR.

El ATTR tiene otros usos en el **Spectrum** aparte de evitar que un carácter entre en una zona prohibida. Por ejemplo, en un laberinto más grande, podrías utilizar el mismo principio para que el carácter haga explosión o se queme si entra en la zona prohibida. Para un laberinto que tuviera INK rojo y PAPER rojo, necesitarías una línea que empezara así:

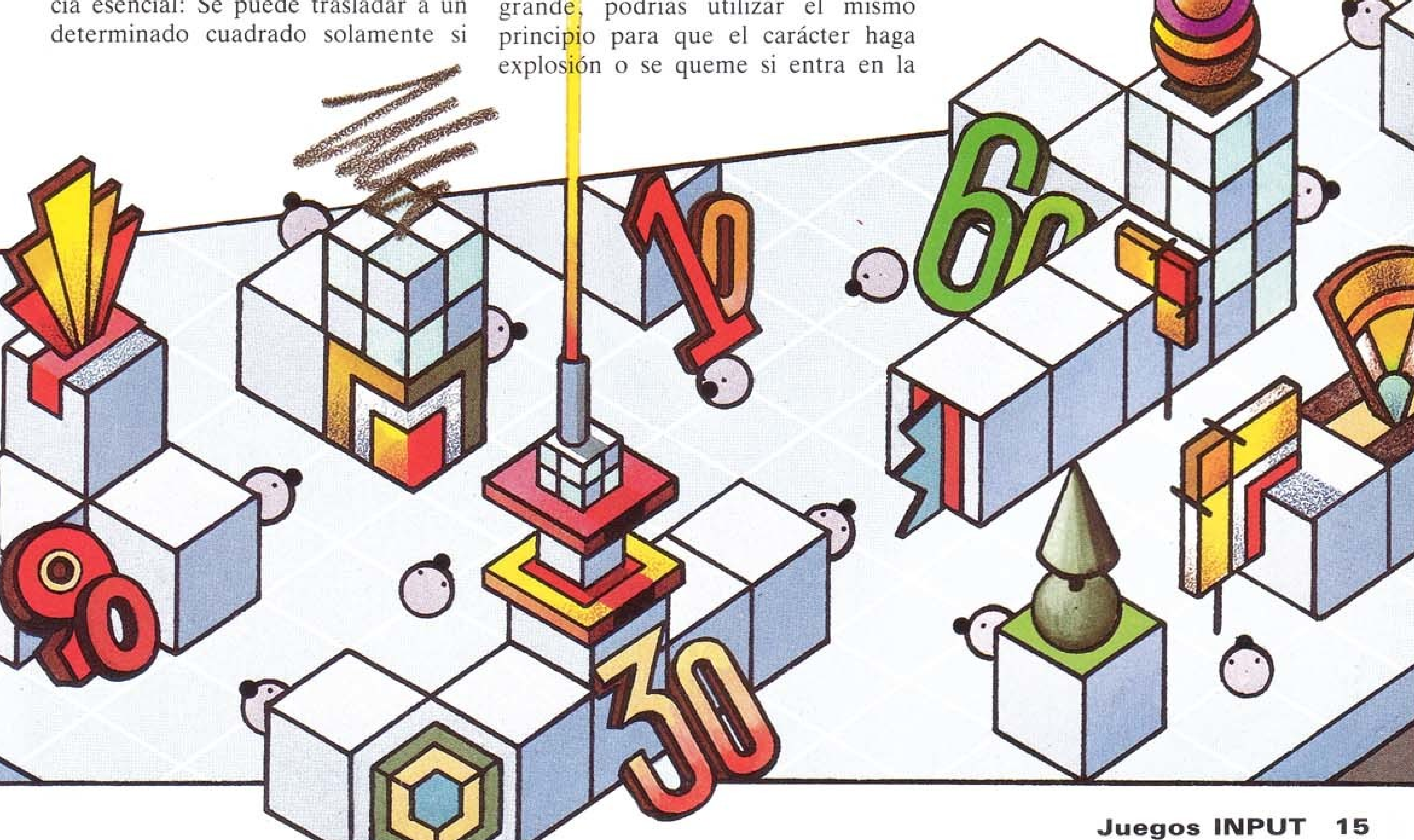
```
IF ATTR (x,y)=18 THEN...
```

Mientras tanto, las líneas 1010, 1020 y 1080 definen la posición que tu carácter acaba de abandonar y borran los puntos que va "comiendo", poniendo espacios con la sentencia PRINT. Estas líneas son útiles en cualquier juego, pero fíjate bien dónde van, son las primeras líneas justo antes de la "acción" de INKEY$ y la que viene inmediatamente detrás.

Observa que la línea 1090 es sólo temporal y que, posteriormente, se prescindirá de ella. Se ha incluido para que puedas ejecutar el programa y probarlo; sin el bucle que se cierra con esta línea, el carácter sólo avanzaría una casilla.

PUNTUACION Y CRONOMETRAJE

Ya tienes todos los elementos para construir un juego sencillo. Para que



de verdad se pueda jugar con él, y en ausencia de "enemigos" —que harían el programa desmedidamente largo— lo mejor es incluir una rutina de cronometraje y tanteo. Agrega pues las siguientes líneas:

```
  10 LET bt=100000
  40 LET s=0
 990 POKE 23672, 0: POKE
     23673, 0
1025 IF S=110 THEN GOTO 2000
1090 IF ATTR (x,y)<>63 THEN
     LET s=s+1: BEEP .005,10
2000 LET t=(PEEK
     23672+256*PEEK
     23673)/50
2010 PRINT AT
     1,6;t;" SEGUNDOS"
2020 IF t<bt THEN LET bt=t
2030 PRINT AT 19,4;"MEJOR
     TIEMPO "; bt;
     " SEGUNDOS"
```

Para probar este fragmento de programa tienes que añadirle temporalmente una línea. Acuérdate de suprimirla después:

```
1100 GOTO 1000
```

El sistema de puntuación es bastante directo. La línea 40 pone a cero el valor inicial de la puntuación. La línea 1090 incrementa la puntuación en 1 cada vez que tu carácter se come un punto, o más correctamente, cada vez que atraviesa una casilla cuyos valores de INK y PAPER no son ambos blancos; como ves, aquí tenemos otra vez a ATTR. La línea 1025, que entra en acción cuando se han comido todos los puntos, pide al ordenador que compruebe (en la línea 2000) cuánto ha tardado el jugador.

La parte de cronometraje es algo más difícil de entender, ya que requiere el conocimiento de los PEEK y POKE, que examinaremos en su momento. Pero, diciéndolo brevemente, la línea 990 pone a cero el "reloj" del **Spectrum**, cargando con ceros las direcciones de memoria adecuadas. La línea 2000 cuenta el número de imágenes visualizados en la pantalla del monitor desde que el juego empezó, y a continuación divide por 50 el número de imágenes por segundo en nuestro televisor.

Al mismo tiempo, la línea 10 sitúa el "mejor tiempo" inicial a 100.000, mucho más de lo que se espera que tarde cualquiera en ser comido, mientras que la línea 2020 compara el último tiempo obtenido con el "mejor tiempo".

¿OTRA VEZ?

Para darle al jugador otra oportunidad, tienes que introducir las siguientes líneas. Teclea primero CAPS y BREAK, y despues pulsa ENTER.

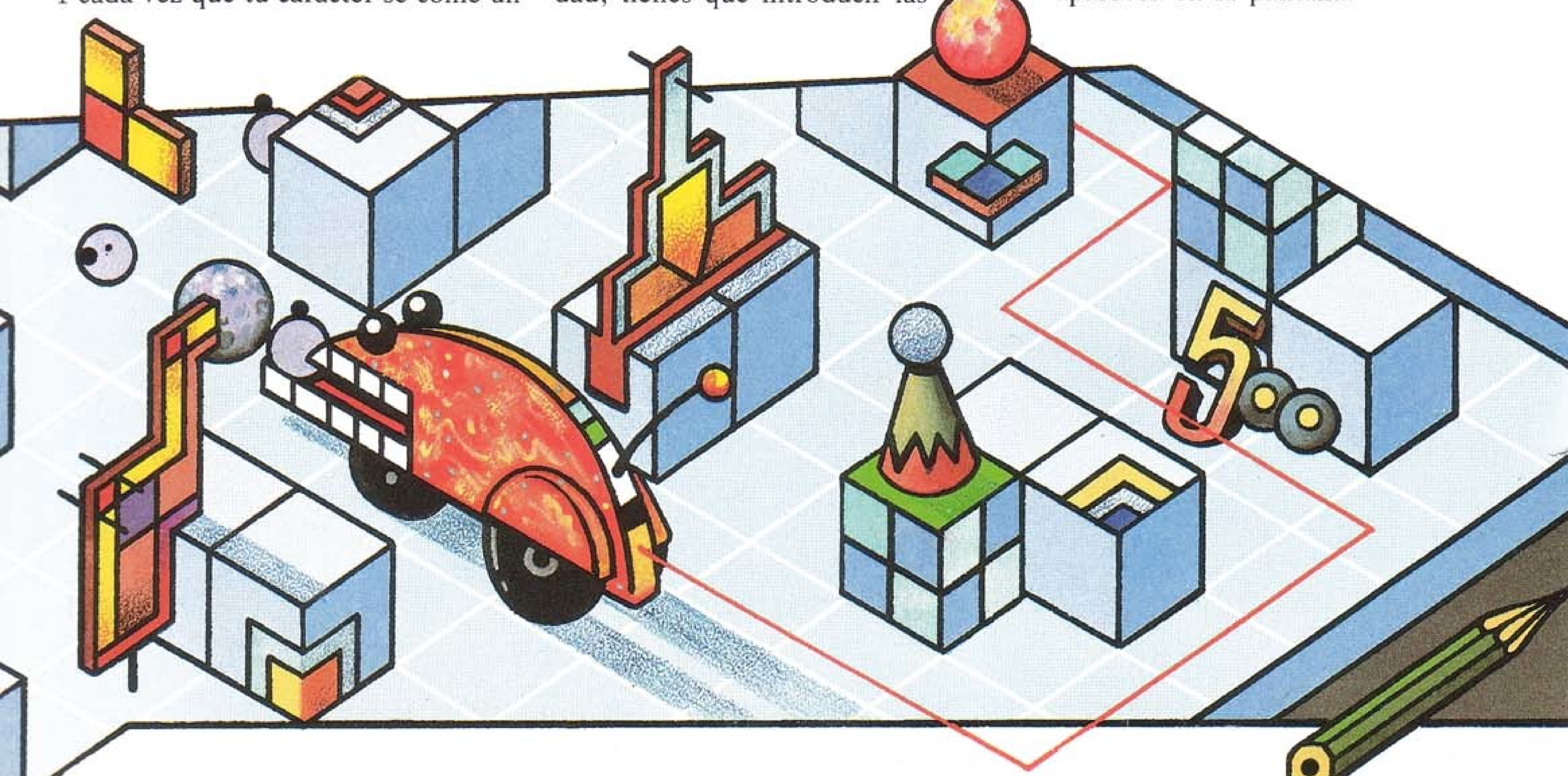
```
2040 PRINT AT 21,2;"PULSA
     UNA TECLA PARA JUGAR"
2050 IF INKEY$<>""THEN GOTO
     2050
2060 IF INKEY$=""THEN GOTO
     2060
2070 RESTORE
2080 GOTO 40
```

Con RESTORE se vuelve al principio de la lista de los DATA.

OTROS LABERINTOS

Si quieres probar otros laberintos con las mismas dimensiones que éste, puedes hacerlo simplemente cambiando la disposición de las letras p en las sentencias DATA, desde la línea 9000 en adelante. Para tener un laberinto más grande o uno más pequeño, también tienes que volver a definir los límites o contornos, cambiando los números que aparecen en las líneas 100 y 120.

Si haces esto, asegúrate de que tienes bastantes datos para llenar todo el laberinto, ya que, de lo contrario, tendrás un mensaje de error, incluso antes de que el laberinto empiece a aparecer en la pantalla.





MICROLID: Gregorio Fdez, 6
Tel.: (983) 35 26 27 VALLADOLID.
BYTE: Plaza Padre Damián, 2
Tel.: (967) 23 78 55 ALBACETE

JORGE JUAN, 116 - 28028 MADRID. TEL.: (91) 274 53 80



SPECTRUM PLUS + CINTAS	**29.800** ptas.
TECLADO DK'TRONICS + 4 PROGRAMAS	**8.990** ptas.
AMPLIFICADOR DE SONIDO	**2.695** ptas.
INTERFACE INDESCOMP PARAL/RS-232	**8.995** ptas.
IMPRESORA GP-50S	**19.990** ptas.
INTERFACE-1	**10.990** ptas.
AMSTRAD 464 (MONITOR + CINTAS)	**57.900** ptas.
AMSTRAD 664 (MONITOR + DISCOS)	**93.900** ptas.
MICRODRIVE	**10.900** ptas.





Si nos pides una de estas novedades, **GRATIS** recibirás 1 COMMUTADOR TV/ORDENADOR, o 1 BOLIGRAFO CON RELOJ INCORPORADO, O 1 LIBRO DE BASIC *(SI, COMPLETAMENTE GRATIS).*

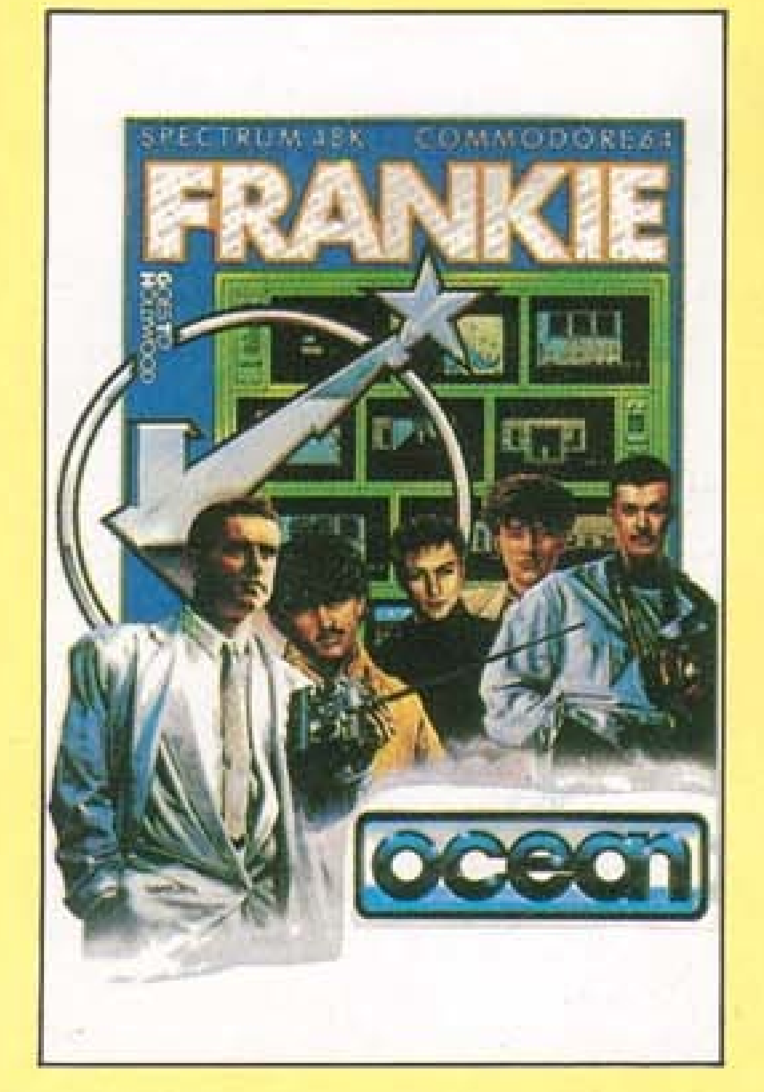
1	______	**1.975**	1	______	**1.795**	1	______	**2.395**
2	______	**2.095**	2	______	**1.995**	2	______	**2.095**

¡¡OFERTAS JOYSTICKS!!

QUICJ SHOT I + INTERFACE T. KEMPSTON	**3.395** ptas.
QUICK SHOT II + INTERFACE T. KEMPSTON	**3.995** ptas.

LAPIZ OPTICO	**3.680** ptas.
Cartucho microdrive	**495** ptas.
Cinta C-15 (especial)	**85** ptas.

SPY HUNTER	**1.975** ptas.
BRUCE LEE	**1.925** ptas.
AIRWOLF	**1.695** ptas.
POLE POSITION	**1.975** ptas.
UNDERWULDE	**1.875** ptas.
ALIEN 8	**1.875** ptas.



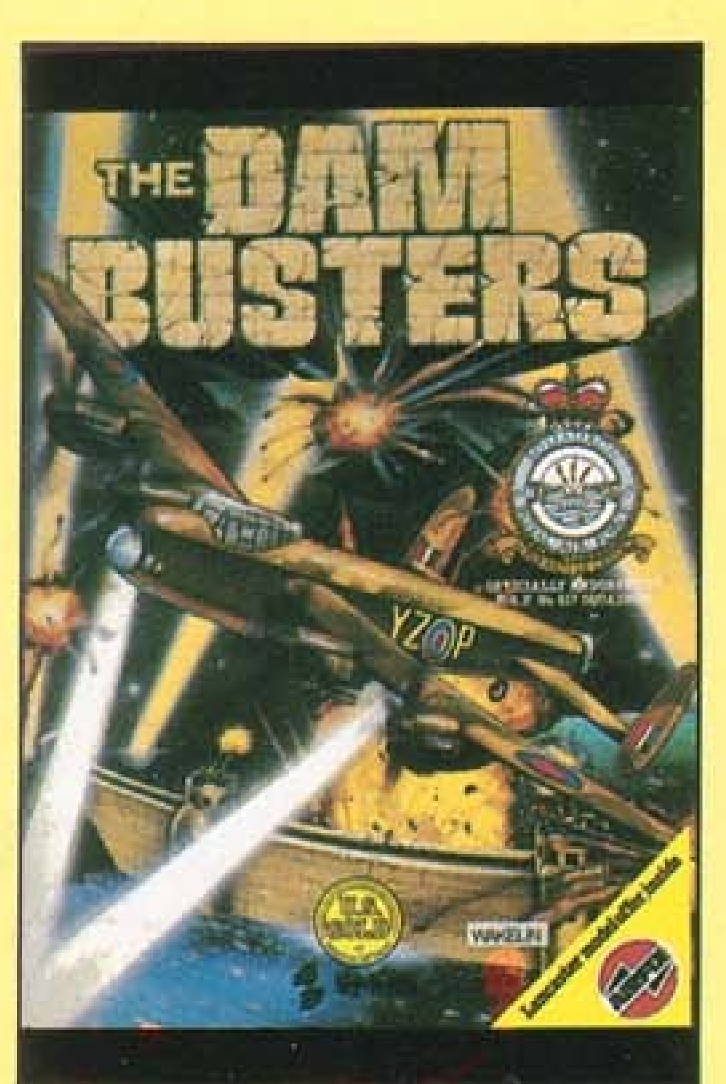




**** 20 por 100 de descuento en todas las impresoras.***

Te recordamos que puedes pedir posters de tus juegos favoritos, adhesivos, bolígrafos o información en general, todo gratuitamente.

Si el pedido lo deseas contra-reembolso *(sin gastos de envío)*, llama al tel. **(91) 274 53 80 o 233 07 81** *(será más rápido el envío)* o escribiendo a **C/ Jorge Juan, 116. 28028 MADRID.**





EL LENGUAJE DE LA TORTUGA (II)

- EDICION
- DISEÑO DE PROCEDIMIENTOS
- COMPROBACION DE LA MEMORIA
- DIBUJANDO UN HEXAGONO

Estrellas, círculos, espirales y hexágonos, mágicamente dibujados en tamaños continuamente crecientes. Son sólo algunas de las formas que puedes conseguir con unas cuantas instrucciones utilizando el LOGO.

En la primera parte de este artículo vimos cómo dibujar con la tortuga del LOGO, cómo crear una primitiva de LOGO enseñando a la tortuga a hacer un dibujo, y cómo utilizar estas primitivas para ayudar a definir nuevas primitivas. Por ejemplo, es posible enseñar a la tortuga a dibujar un hexágono:

```
TO HEXAGONO
REPEAT 6 [FORWARD 70 RIGHT
60]
END
```

Después de teclear END, el LOGO te presentará el mensaje HEXAGONO DEFINED para que sepas que, a partir de ahora, HEXAGONO forma parte de su vocabulario. Cada vez que teclees HEXAGONO, la tortuga dibujará un hexágono y terminará apuntando en la misma dirección que tenía cuando empezó:

Puedes utilizar HEXAGONO para definir MODELO:

```
TO MODELO
REPEAT 12 [HEXAGONO FORWARD
10 RIGHT 30]
END
```

Será bastante raro que tus procedimientos de LOGO hagan lo que tú esperas a la primera. Lo normal es que haya por lo menos un error escondido por alguna parte del programa. Por ejemplo, al dibujar el hexágono podrías haber supuesto que, dado que el ángulo entre los lados contiguos es de 120 grados, tendrías que decir a la tortuga REPEAT 6 [FORWARD 60 RIGHT 120]. La figura resultante no habría sido un hexágono.

EL CAMPO VISUAL DE LA TORTUGA

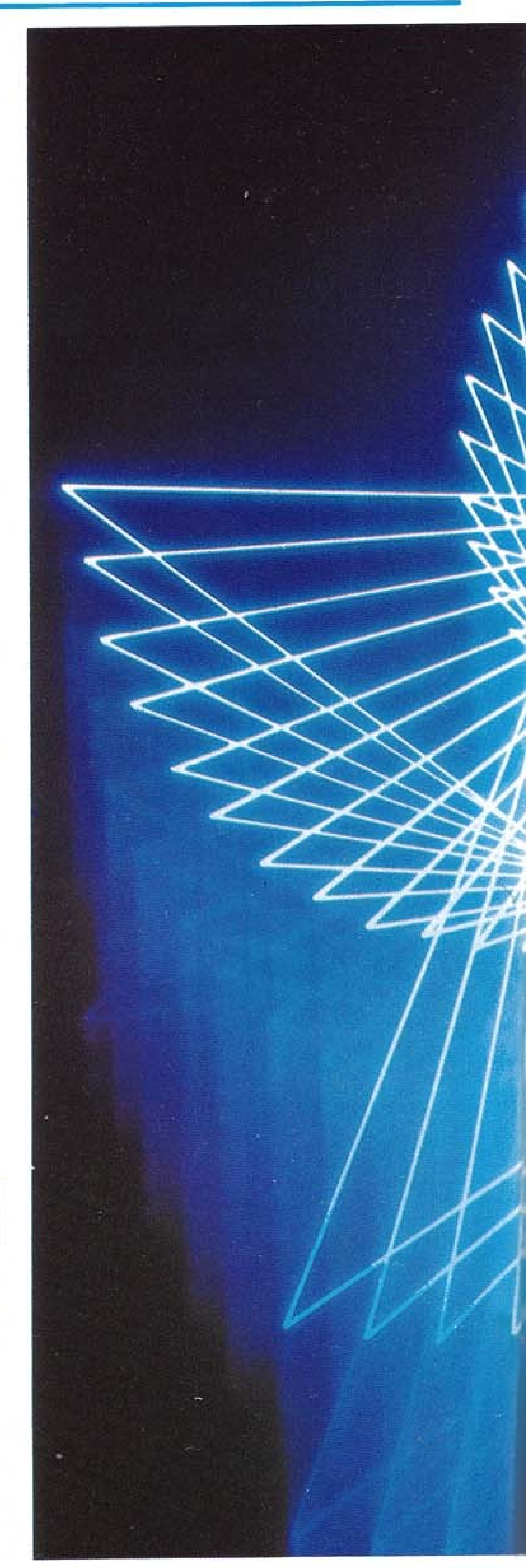
La geometría de la tortuga es diferente a la geometría de coordenadas, que, probablemente, te resulta más familiar, donde las posiciones se definen con referencia a un punto externo. (Las coordenadas normales de la pantalla en los gráficos de tu ordenador se definen de esta forma.) La geometría de la tortuga funciona de manera diferente, ya que el sistema de referencia está en la propia tortuga. Al dibujar un hexágono, la tortuga gira 60 grados y no 120, aunque las dos líneas que la tortuga ha dibujado formen entre sí un ángulo de 120 grados. Si tienes algún problema para entender el sistema de referencia de la tortuga, imagínate que vas andando por la figura que quieres dibujar. Los ángulos y distancias que tienes que describir y recorrer son los que has de traducir en instrucciones para la tortuga.

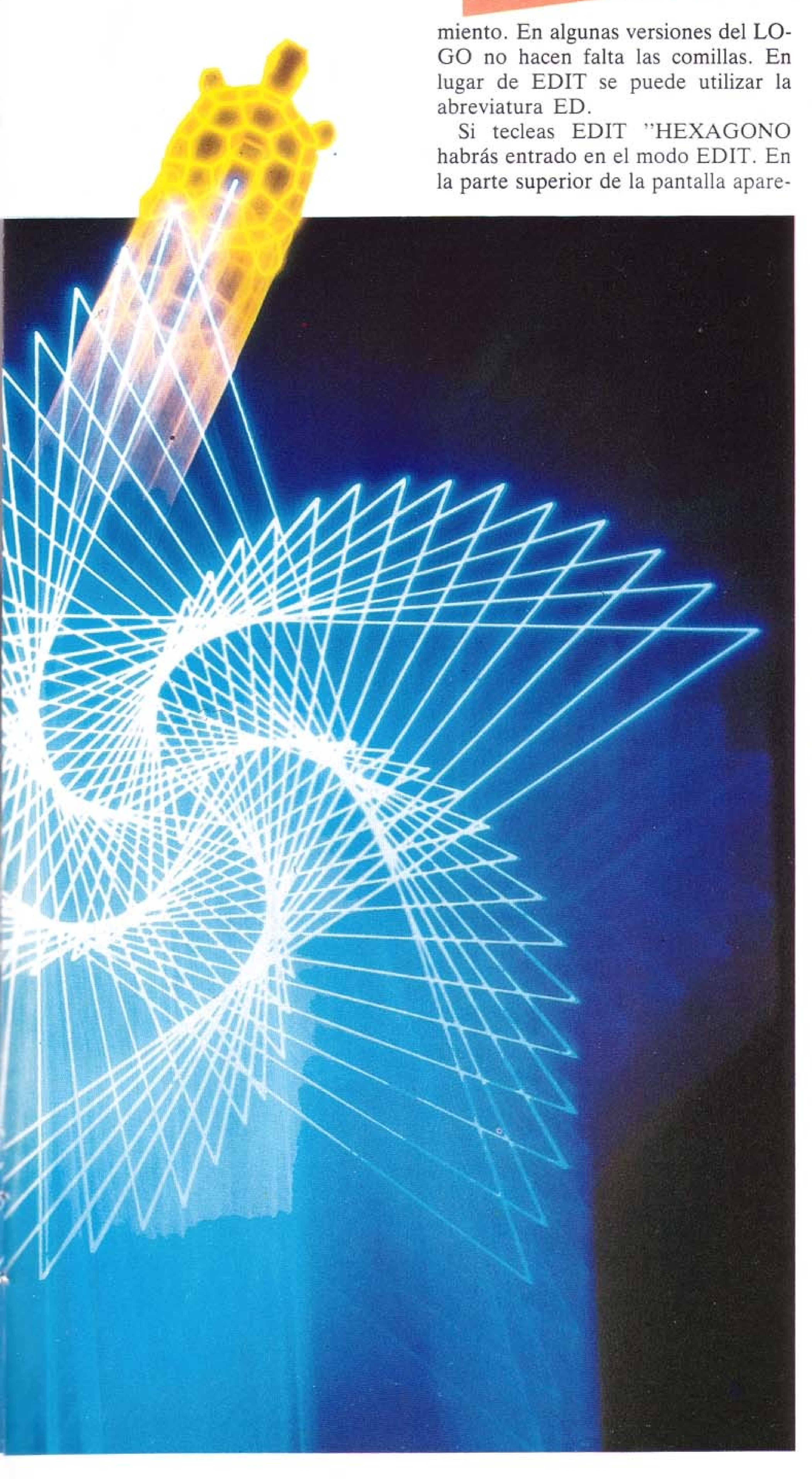
Aunque esto puede parecer confuso, cuando te hayas acostumbrado te resultará realmente fácil, como suele suceder a los niños, ya que se relaciona directamente con su propia experiencia.

EDICION

Cuando los errores de programación empiezan a hacer estragos, hay que eliminarlos del programa. Para poder hacer esto, hay que ir a un tercer modo, el *Editor*. El modo de edición es semejante al modo de procedimiento. No afecta al estado inmediato de la tortuga.

Para editar un procedimiento, teclea EDIT" y el nombre del procedi-





miento. En algunas versiones del LOGO no hacen falta las comillas. En lugar de EDIT se puede utilizar la abreviatura ED.

Si tecleas EDIT "HEXAGONO habrás entrado en el modo EDIT. En la parte superior de la pantalla aparecerá la definición de HEXAGONO. En el modo EDIT no hay gráficos, toda la pantalla está dedicada al texto. Utilizando las teclas de desplazamiento del cursor, puedes mover éste hacia arriba o hacia abajo de una línea de programa a otra, o hacia delante o hacia atrás dentro de cada línea.

El caracter que en cada momento está a la izquierda del cursor se puede suprimir, igual que ocurría al editar una línea individual en el modo procedimiento. El caracter que hay en el mismo lugar del cursor puede suprimirse con la tecla de borrado con CAPS SHIFT. Para insertar caracteres no hay más que teclearlos, el resto de la línea ya se irá moviendo para hacerles sitio.

En el modo EDIT se puede definir un nuevo procedimiento. La ventaja de hacer esto es que puedes cambiarte de una línea a otra para modificar instrucciones mientras todavía estás definiendo el procedimiento. Para definir Flor de esta manera, tienes que teclear:

```
EDIT "FLOR
```

En la parte superior de una pantalla vacía aparecerá TO FLOR. Puedes definir ahora el procedimiento de la forma corriente, aprovechando las ventajas que te brinda el eitor de pantalla.

Cuando se define un procedimiento, el modo de edición te será accesible tanto si teclea TO como si tecleas EDIT.

DALE LAS ENTRADAS CORRECTAS

Existe una importante diferencia entre HEXAGONO y RIGHT. Con HEXAGONO siempre dibujará un hexágono de 60 unidades de lado. En cambio, RIGHT requiere un valor de entrada que diga a la tortuga cuánto tiene que girar a la derecha; de la misma forma, también FORWARD, BACK y LEFT requieren argumentos o valores de entrada.

Se puede volver a definir HEXAGONO para que también requiera argumentos de entrada. Para hacer esto

asignamos un nombre a la entrada y lo incluimos en la línea que contiene el título del procedimiento. El nombre de la entrada ha de ir siempre precedido por dos puntos. El nombre de la entrada y los dos puntos se utilizan en la definición del procedimiento, siempre que el valor de la entrada aparezca normalmente. Esto se entenderá mejor con un ejemplo.

Si al argumento de entrada para HEXAGONO le llamas LADO, al teclear EDIT "HEXAGONO estarás en modo editor. En la línea que contiene el título, añade ahora el nombre del argumento de entrada precedido por dos puntos, de forma que se lea:

```
TO HEXAGONO :LADO
```

En la siguiente línea suprime el 70 y teclea LADO en su lugar. La definición de hexágono queda ahora así:

```
TO HEXAGONO :LADO
REPEAT [6 FORWARD :LADO
RIGHT 6Ø]
END
```

Si ahora tecleas HEXAGONO, el LOGO te responderá con:

```
NOT ENOUGH INPUTS TO
HEXAGONO (faltan entradas
para HEXAGONO)
```

El trato que recibe HEXAGONO es el de cualquier otra primitiva del LOGO que requiere una entrada. Si tecleas HEXAGONO 20, la tortuga dibujará un hexágono de 20 unidades de lado. Con HEXAGONO 100 dibujará un hexágono de 100 unidades de lado, y así sucesivamente.

MODELO debería ser nuevamente escrito de la siguiente manera:

```
TO MODELO
REPEAT 12 [HEXAGONO 7Ø
FORWARD 1Ø RIGHT 3Ø]
END
```

Puedes modificar MODELO para tomar LADO como entrada.

```
TO MODELO :LADO
REPEAT 12 [HEXAGONO :LADO
FORWARD 1Ø RIGHT 3Ø]
END
```

Utiliza este mismo principio en CIRCULO para hacer que la tortuga dibuje círculos de diferentes tamaños.

```
TO CIRCULO :MEDIDA
REPEAT 72 [FORWARD :MEDIDA
RIGHT 5
END
```

Realmente, la tortuga está dibujando un polígono de 72 lados en lugar de un verdadero círculo, aunque el resultado es una curva razonablemente regular. La tortuga dibuja los círculos bastante lentamente, pero puedes agilizar las cosas haciendo que la tortuga sea invisible; de esta forma, el LOGO no tendrá que dibujar a cada paso. Para hacer esto, utiliza HIDERTURTLE, que se puede escribir abreviadamente como HT.

Cuando quieras que la tortuga sea otra vez visible, utiliza SHOWTURTLE, abreviado ST. Ahora puedes modificar CIRCULO para que sea más rápido, así:

```
TO CIRCULO :MEDIDA
HIDETURTLE
REPEAT 72 [FORWARD :MEDIDA
RIGHT 5]
SHOWTURTLE
END
```

PONLO EN UN FICHERO

Una vez que un procedimiento ha sido definido, pasa a formar parte del vocabulario del LOGO, pero cuando apagues el ordenador, todas tus primitivas desaparecerán de la memoria. Con el LOGO puedes guardar tu trabajo, en disco o en cinta, utilizando SAVE seguido de ". Si se te olvidan las comillas, el LOGO no guardará nada y todos tus esfuerzos habrán sido en vano.

Al utilizar SAVE, el LOGO pone todo el contenido de tu "espacio de trabajo" —todos los procedimientos que hayas definido— en un disco o *cassette*. SAVE "TRABAJO guardaría todos los procedimientos que haya en la memoria del ordenador en un disco o cinta, creando un fichero llamado TRABAJO.

Después de cargar el LOGO, puedes recuperar el fichero del disco o de la cinta con LOAD (en algunos casos, con READ). LOAD "TRABAJO tomará todos los procedimientos que haya en el fichero llamado TRABAJO y los cargará en el área de trabajo del ordenador.

Si en la memoria del ordenador hay un procedimiento con un determinado nombre, y cargas otro procedimiento con el mismo nombre, el procedimiento recién cargado sustituirá al que ya estaba allí, el cual se perderá sin dejar el menor rastro.

Los ficheros que hay en el disco pueden borrarse del mismo con ERASFILE " y el nombre del fichero. A veces se utiliza ERS en vez de ERASEFILE. Así, ERASEFILE "TRABAJO borrará todos los procedimientos que estén el fichero TRABAJO del disco. No se puede utilizar ERASEFILE con un *cassette*. Para borrar un fichero de una cinta tienes que guardar otra área de trabajo en el sitio del fichero que quieres borrar.

El uso de ERASFILE no afecta a los procedimientos que están en la memoria del ordenador, sino sólo a los que ya han sido guardados en disco.

Como el LOGO guarda los procedimientos de la memoria sobre la base de "todo o nada", es conveniente ordenar el contenido de la misma antes de proceder a su almacenamiento en disco. Existen varias primitivas del LOGO relacionadas con la gestión del área de trabajo.

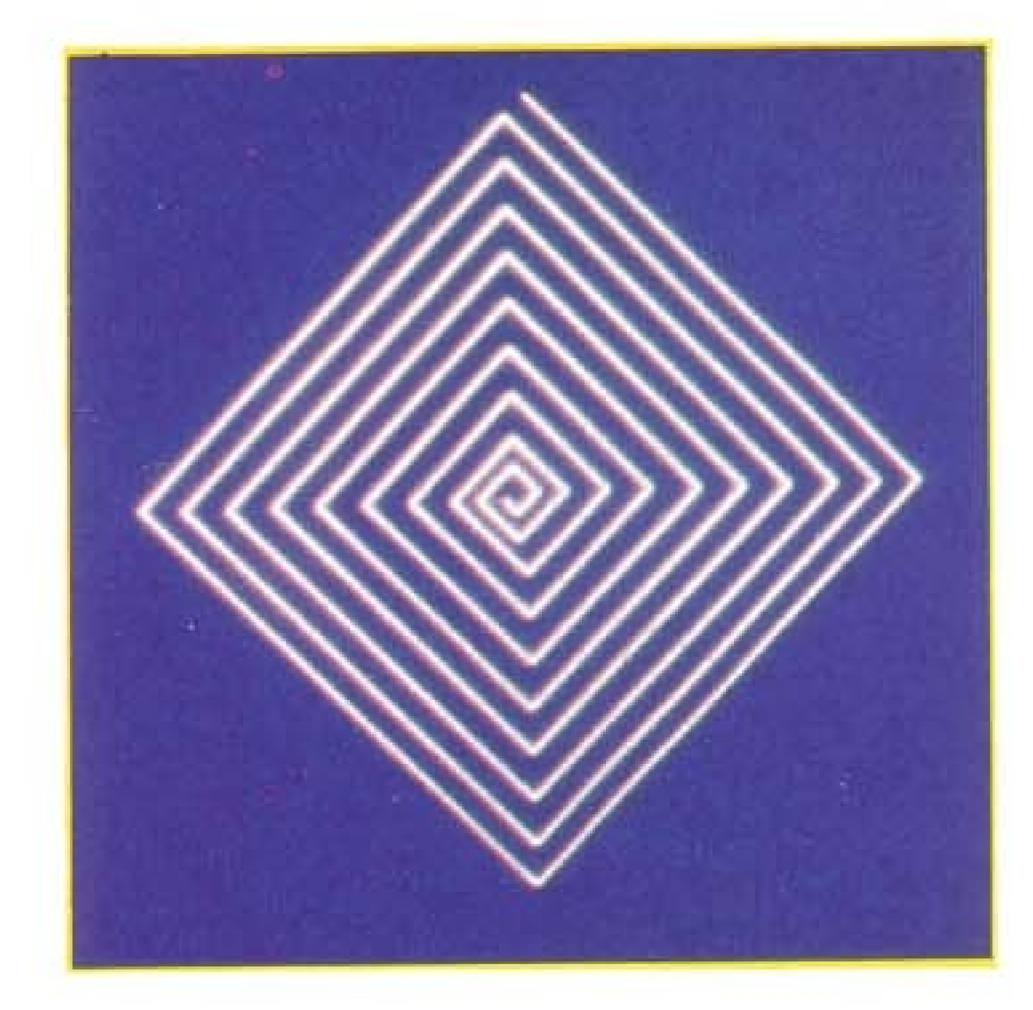
COMPROBACION DE LA MEMORIA

El área de trabajo de la memoria es una lista de nudos. Cada nudo consta de cinco bytes. Después de cargar el LOGO hay que manejar unos 3.000 nudos. Se utilizan al definir o cargar procedimientos. Para ver qué procedimientos hay en tu ordenador, utiliza POTS. Esto quiere decir *Print out titles* (escribe los títulos), escribiéndose los títulos de todos los procedi-

mientos que hay en el área de trabajo. Por ejemplo:

```
POTS
TO HEXAGONO :LADO
TO MODELO :LADO
TO FLOR
TO CIRCULO :MEDIDA
```

PO quiere decir imprimir *(Print Out)*. PO seguido del nombre de un procedimiento imprime la definición del mismo. Por ejemplo:



```
TO CIRCULO
TO CIRCULO :MEDIDA
HIDETURTLE
REPEAT 72 [FORWARD :MEDIDA
RIGHT 5]
SHOWTURTLE
END
```

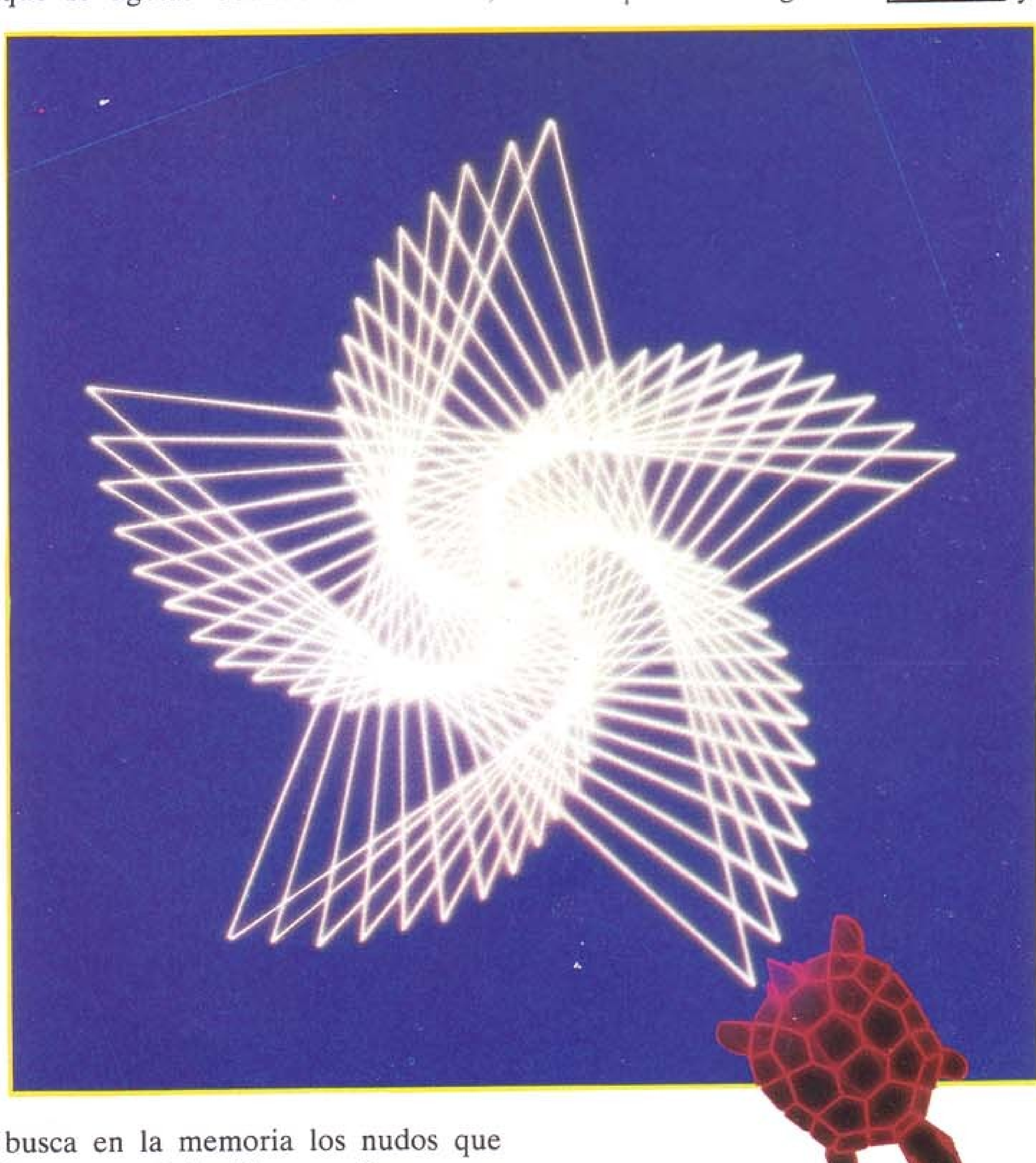
POALL significa imprimir todo. Imprime la definición de todos los procedimientos que hay en el área de trabajo. Después de ver qué procedimientos hay en la zona de trabajo puede ser que quieras borrar algunos de ellos. ERALL significa borrar todo *(Erase all)*, y borra todos los procedimientos que hay en la memoria.

Para borrar procedimientos aislados o en grupos utiliza ERASE. Para borrar un solo procedimiento, pon comillas delante del nombre de dicho procedimiento. Por ejemplo: ERASE "FLOR eliminará del área de trabajo el procedimiento FLOR.

Para borrar un grupo de procedimientos tienen que encerrar sus nombres entre paréntesis cuadrados. Por ejemplo: ERASE [FLOR CIRCULO MODELO] actuará borrando dichos procedimientos de la memoria del ordenador.

Con CATALOG obtendrás una lista de los ficheros almacenados en el disco que estás usando.

Al borrar procedimientos del área de trabajo vuelven a quedar nudos listos para ser utilizados de nuevo. No vuelven a añadirse a la lista de nudos libres del LOGO. Este continúa trabajando con su lista original hasta que se agota. Cuando esto ocurre,



busca en la memoria los nudos que hayan quedado libres y forma con ellos una nueva lista. A esto se le llama "recogida de basura"» y a veces hace que el LOGO se pare durante un segundo o dos.

VUELTAS Y MAS VUELTAS

Volvamos de nuevo al tablero de dibujo. He aquí otra forma de dibujar el hexágono.

```
HEXAGONO 2 :LADO
END
```

Si lo pruebas, te quedarás confuso y aturdido. Dibuja un hexágono de un tamaño dado, igual que hacías HEXAGONO. Pero la tortuga se comporta de una forma muy diferente. En HEXAGONO, la tortuga se detenía cuando había terminado el dibujo. En HEXAGONO2, la tortuga sigue dando vueltas y más vueltas hasta que la detengas con BREAK y la tecla G. Esto sucede porque si, por ejemplo, tecleas HEXAGONO50, la tortuga ejecuta FORWARD50 RIGHT60 y entonces ha de hacer HEXAGONO2 50 una vez, y otra, y otra, hasta que la mandes parar.

Igual que un procedimiento puede llamar a otro procedimiento como parte de su definición, también puede llamarse a sí mismo como parte de sí

mismo. Este introvertido estado de cosas se llama recursión.

Un procedimiento puede tener más de una entrada. Aquí tienes un ejemplo sencillo:

```
TO RECTANGULO :LADO1 :LADO2
REPEAT 2 [FORWARD :LADO1
RIGHT 90 FORWARD :LADO2
RIGHT 90]
END
```

Así RECTANGULO20 40 dibujará un rectángulo de lados 20 y 40 unidades.

La entrada puede ser un ángulo o un lado.

```
TO MOVIMIENTO :LADO :ANGULO
REPEAT 60 [FORWARD :LADO
RIGHT :ANGULO]
END
```

MOVIMIENTO toma una entrada de su paso FORWARD y otro de la amplitud de su giro.

Un procedimiento muy corriente en LOGO, parecido a MOVIMIENTO, es POLY.

```
TO POLY :MEDIDA :ANGULO
FORWARD :MEDIDA
RIGHT :ANGULO
POLY :MEDIDA :ANGULO
END
```

POLY es un sencillo procedimiento recursivo que puede producir algunos dibujos atractivos.

Igual que HEXAGONO2, POLY pondrá a la tortuga a realizar un trabajo que no termina nunca. Tienes que detenerla interrumpiendo el programa.

Un desarrollo interesante de POLY, que va más allá de la mera repetición de las mismas instrucciones es ESPIRAL.

```
TO ESPIRAL :LADO :ANGULO
FORWARD :LADO
RIGHT :ANGULO
ESPIRAL :LADO+2 :ANGULO
END
```

Como el LOGO permite el uso de funciones matemáticas, en la cuarta línea se suma 2 al valor que teníamos de LADO.

Al teclear ESPIRAL 172 la tortuga ejecutará la siguiente secuencia:

```
FORWARD 1
RIGHT 72
FORWARD 3
RIGHT 72
FORWARD 5
RIGHT 72
FORWARD 7
RIGHT 72
FORWARD 9
```

y así sucesivamente. Esto va produciendo una espiral pentagonal. Cada lado de la espiral es dos pasos más largo que el lado anterior.

¡ALTO!

No hay manera de detener a la tortuga como no sea interrumpiendo el programa, pero puedes imponer una condición dentro del procedimiento, para que la tortuga se detenga cuando el lado alcance una determinada longitud.

STOP es una primitiva del LOGO que sólo puede utilizarse dentro de un procedimiento. No se puede utilizar de modo inmediato o a "alto nivel" cuando el programador tiene el control y el LOGO está esperando una instrucción. Detiene el procedimiento antes de que se llegue a la sentencia final y devuelve el control al usuario. El usuario podría ser otro procedimiento del LOGO. Si no hay otro procedimiento, el control vuelve al usuario de verdad a alto nivel. STOP solamente detiene el procedimiento en el que aparece.

También puedes imponer una condición con IF y THEN. En algunas versiones no se requiere THEN. IF examina alguna condición y decide si es cierta o no. Si es cierta, se pone en movimiento una determinada acción. Si la condición no es cierta, se pone en marcha otra acción diferente.

Con IF se prueba una condición dentro de un procedimiento; si no es cierta, se ejecuta la siguiente línea del programa. Si es cierta, THEN dice al LOGO lo que tiene que hacer.

En el siguiente procedimiento puedes ver en acción a STOP IF y THEN.

En el procedimiento ESPIRAL, sustituye LADO por HEXAGONO, con lo que la tortuga dibujará una espiral de hexágonos cada vez más grandes. Los condicionales detendrán el procedimiento cuando los lados del hexágono alcancen un total de 20 unidades.

```
TO GIROHEX :LADO :ANGULO
IF :LADO>100 THEN STOP
HEXAGONO :LADO
RIGHT :ANGULO
GIROHEX :LADO+3 :ANGULO
END
```

El LOGO interpreta el signo > con el significado de "mayor que", del mismo modo que reconoció la función matemática de +.

GIROHEX 110 hace que la tortuga dibuje un hexágono de lado 1 unidad, gire 10 grados, dibuje un hexágono de lado 4 unidades, gire 10 grados, dibuje un hexágono de lado 7 unidades, y así sucesivamente, aumentando cada vez el lado del hexágono en 3 unidades. La segunda línea comprueba cada vez si los lados son mayores que 100. Si no es así, pasa a la línea siguiente y dibuja un nuevo hexágono. Cuando el lado se hace mayor que 100, entra en acción THEN. Le dice al LOGO que se pare (STOP) y el procedimiento se detiene.

En estos artículos hemos cubierto ya bastante material geométrico como para entretener durante un año a un chaval de diez años. La recursión hace posible obtener una amplia variedad de resultados a partir de programas sencillos. Con el editor de pantalla se pueden corregir los procedimientos y se estimula la experimentación. Los resultados de los programas son con frecuencia agradablemente sorprendentes. La tortuga convierte las instrucciones en imágenes y hace más fácil ver dónde están los errores de programación.

En la última parte de este artículo veremos los duendes del LOGO, el proceso de textos y de listas, y las posibilidades matemáticas.



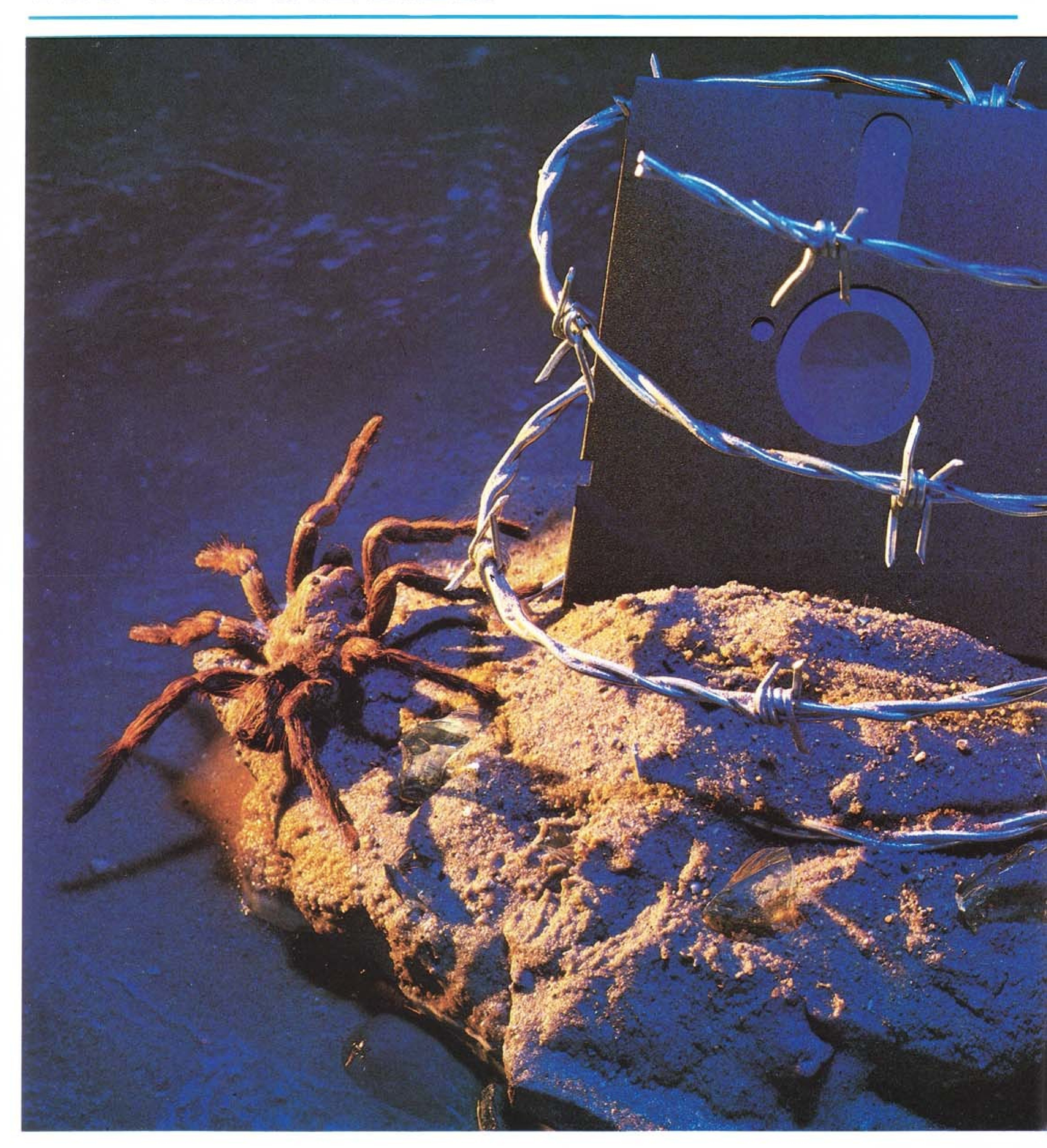
LOS MEJORES DE INPUT SINCLAIR

PUESTO	TITULO	PORCENTAJE
1.º	Profanation	21,7 %
2.º	Skool Daze	18,3 %
3.º	Rocky	11,01 %
4.º	Knight Lore	7,9 %
5.º	Alien 8	9,03 %
6.º	Panorama para matar	8,4 %
7.º	Hyper Sports	8,05 %
8.º	Every one's a wally	6,4 %
9.º	Decathlon	5,3 %
10.º	Bruce Lee	3,91 %
		100 %

Octubre de 1985

PROTEGE TUS PROGRAMAS

- LO QUE PUEDES HACER
- INTERDEPENDENCIA DE PROGRAMAS
- AUTOEJECUCION DE UN PROGRAMA



¿Quieres proteger tus técnicas especiales contra miradas indiscretas? ¿Tal vez desees añadir determinados toques profesionales a tu programa? A continuación te presentamos lo que el BASIC puede ofrecerte.



Siempre es una buena cosa proporcionarles a tus programas una apariencia profesional, una vez ajustadas todas las piezas. Siempre es posible mejorar su presentación, pero también puedes dotarle de una protección razonable para que no sean abiertos fácilmente.

La protección de los programas en BASIC recae en la inclusión de estrategias disuasorias, programas escritos total o mayoritariamente en código máquina, que pueden hacer uso de técnicas de protección más sofisticadas. Sin embargo, las empleadas para los programas en BASIC no pueden ser aplicadas a los programas en código máquina.

Para prevenir la copia por parte de los «piratas» se pueden emplear todo tipo de trucos, aunque estos sólo deseen LISTar el programa por curiosidad. Cuantos más decidas integrar en tus programas, tanto mejor para tí. En otros programas no merece la pena el esfuerzo.

De una cosa puedes estar absolutamente seguro a este respecto: No hay forma de proteger a un programa que lo haga completamente seguro contra la copia. Mucha gente se toma lo de abrir programas como uno más de los desafíos de la vida. Otros intentan entrar en el programa para examinarlo, aprender o simplemente modificar una rutina que le permita ajustarlo a sus necesidades.

PRIMEROS PASOS

Un método de proteger a los programas se centra en proporcionarles muchas trampas sencillas. Con ellas no se derrotará a quienes tengan algún conocimiento sobre la máquina, pero será una laboriosa tarea a realizar. Desafortunadamente, algo como esto puede convertirse en bastante tedioso para el escritor de programas, que deberá preocuparse de la protección al mismo tiempo que escribe el programa. El resultado podría traducirse en mayores complicaciones a la hora de depurar el programa.

Los cierres de este tipo hacen poco más que introducir cambios que hagan imposible el SAVE y LIST normales y dificultan algunos otros comandos relacionados con la posterior edición del programa.

CARGADORES

Una protección que resulta algo más ventajosa es disponer un mecanismo de autoejecución del programa una vez que se ha cargado (LOAD). Con el **Spectrum** esto se consigue de un modo bastante simple, utilizando los comandos LOAD y SAVE adecuados, como se verá más adelante.

Estos comandos se introducen habitualmente en modo directo durante el proceso normal de carga (LOAD) de un programa. Pero también pueden ser reclamados con igual facilidad por un programa que se carga inmediatamente antes que el programa principal. Tal programa se llama «cargador» y puede estar escrito en BASIC o código máquina, dependiendo de las tareas específicas que deba llevar a cabo. En su forma más simple puede tomar la siguiente apariencia:

```
10 LOAD"NOMBRE DEL PROGRAMA"
```

Ejecutando esta línea (RUN), será cargado automáticamente el programa que lleve el nombre incluido entre las comillas.

Obviamente, una línea sencilla como ésta no tiene mucho sentido. En realidad los cargadores pueden servir para hacer mucho, mucho más. Se les utiliza frecuentemente para hacerse cargo de la programación sumplementaria, responsable de cosas tales como la visualización de la pantalla de presentación, avisos de *copyright*, carga y visualización de instrucciones, ajuste de variables y una gran cantidad de trucos de protección.

Entre los trucos se incluyen algunas de las más potentes formas de proteger un programa en BASIC, conseguidas mediante la manipulación de los comandos propios del sistema en sí. Pero aquí, el principal uso de los cargadores consiste en la puesta en marcha del programa, que permite al código máquina ser ejecutado partiendo del BASIC.

Los cargadores son empleados en muchos tipos de programas comercia-

P y R.

¿Bajo qué circunstancias y en qué áreas de la memoria es posible que pueda almacenar rutinas o llamadas destinadas a incluir rutinas de protección?

Un área no sospechada y «segura» en el **Spectrum** es el *buffer* de la impresora. Está ubicado en las direcciones 23296 a 23551. Esto te proporciona 256 *bytes* de memoria. Por supuesto que en este caso no puedes utilizar el *buffer* cuando el programa situado en la memoria necesite acceder a la impresora.

les. Cantidad de ellos son grabados formando parte de programas compuestos por varias partes, siendo reclamados y cargados posteriormente en respuesta al comando incluido en uno o más cargadores. Recuerda que el BASIC no puede ser llamado normalmente sin reescribir el comando de carga del cargador (LOAD) o, en otras palabras, el programa de carga en sí. Suponiendo que estés tentado por la utilización de dicha técnica, asegúrate de que los módulos escritos en código máquina sean llamados y cargados en primer lugar.

Con los programas compuestos por múltiples partes, los métodos de protección pueden incluso recaer en determinado nivel de interdependencia entre un fichero (parte del programa total) y otro. Lo que ocurre es que un fichero comprueba un valor de localización ajustado por otro. También se podría obtener un programa que compruebe un fichero con datos especiales situado detrás del programa principal. En ambos casos, algo que no estuviera presente haría que el sistema no funcionase o que no se produjera la ejecución del programa.

La mayoría de las veces, un programa de partes múltiples contendrá uno o más módulos escritos totalmente en código máquina.

Una ventaja final de los cargadores son los tiempos de carga más rápidos, posibles gracias a que el código máquina, los datos de la pantalla y los datos sobre caracteres pueden ser depositados directamente en la memoria, en lugar de utilizar sentencias DATA en programas BASIC, que solamente pueden realizar su tarea después de que tecleemos RUN. Todo esto proporciona la impresión de acabado profesional en nuestros programas que, incluso si están desarrollados en BASIC, pueden ser ejecutados sin que parezcan estarlo.

AUTOEJECUCION

¿Cómo puede ser utilizado el cargador para la autoejecución de un programa que le sigue? En el **Spectrum** es simplemente un caso de utilizar el comando adecuado en algún lugar dentro de tu pograma de carga:

```
990 LOAD "NOMBRE DEL
    SIGUIENTE PROGRAMA"
```

Pero recuerda que este programa tendrá que haber sido previamente almacenado empleando el comando de autoejecución SAVE «NOMBRE DEL SIGUIENTE PROGRAMA» LINE 1 o cualquier otro número de línea que represente el comienzo del programa. Si se eligió un número de línea más elevado, puedes incluso añadir algunos avisos de *copyright* o datos en código que pudieran ser PEEKeados para comprobación de seguridad dentro de líneas REM.

UTILIZANDO LAS VARIABLES DEL SISTEMA

Sin embargo, la autoejecución no es suficiente por sí misma. También debes proveer algunos medios que aseguren que el programa no puede ser detenido, LISTado o rectificado.

La forma de conseguirlo es realizar ajustes en la forma en que el sistema reacciona cuando se reclama algo en particular; por ejemplo, la subrutina correspondiente a LIST.

Todo ordenador viene completo con un sistema operativo y, por lo que a nosotros concierne, un intérprete del BASIC. La mayoría de la información que compete al sistema está guardada en memoria de sólo lectura (ROM), pero parte de ella es transferida a la memoria de acceso aleatorio (RAM) cuando el sistema es conectado. Esta es la información que puede ser alterada por el programador para cambiar el modo en que opera el sistema. El resultado produce medidas de seguridad sofisticadas, que veremos en futuros artículos. Como se trabaja con las formas de hacer del ordenador, necesitarás una lista documentada de las variables del sistema y las subrutinas.

Un tipo especial de variable del sistema es el **puntero** o **vector.** Esto consiste normalmente en dos direcciones adyacentes que almacenan la dirección de una subrutina específica. Si cambiamos esta dirección, entonces el sistema operativo es redireccionado a otro lugar siempre que sea reclamada esa subrutina. Todas estas subrutinas del sistema son, naturalmente, código máquina.

Aquellos **punteros** o **vectores** de interés particular —por lo que a nosotros afecta— incluyen a los equivalentes del vector de LIST, el vector de SAVE, los vectores de interrupción y los de inicialización del sistema.

ESTUDIO DE CASOS

Es difícil, si no imposible, proporcionar una protección total contra la piratería, especialmente si intentas no verte profundamente inmerso en el código máquina.

Una de las comprobaciones más simples consiste en insertar una sentencia de *copyright* intocable dentro de tu programa. Esta podría ser enlazada incluso a una rutina que PEEKeará, con el fin de comprobar su presencia, y borrará toda la memoria si detectase algún tipo de trampa.

Primero se calcula la dirección del área destinada al programa en BASIC. Esta dirección (la variable del sistema PROG) está almacenada en las direcciones 23635 y 23636 y pueden ser determinadas así:

PRINT PEEK 23635 + 256*PEEK 23636. Una vez conocido el valor de

PROG, puedes POKEar cualquier número N en (PROG +1) y cambiarás la primera línea de tu programa por la línea N.

El secreto consiste en hacer N igual a cero, debido a que no es posible deshacerse de la línea 0 en un programa. Supón que la primera línea es:

```
10 REM(c) PEPE 1985
```

Escribe el comando directo:

```
POKE(PEEK 23635+256*PEEK
  23636)+1,0
```

Ahí lo tienes: la línea 10 se convierte en la línea 0. Obviamente, esto se podría haber hecho desde un programa cargador.

Por lo que a un programa de autoejecución concierne, la manera obvia de detenerlo es presionar BREAK. Esto dispone un mensaje de BREAK en la parte inferior de la pantalla y permite entonces que el programa sea LISTado.

Pero suponte que la parte inferior de la pantalla no acepte el mensaje. Si observas la lista de variables del sistema que aparece en el manual del **Spectrum,** verás que DF SZ (en la dirección 23659) almacena el número de líneas que hay en la parte inferior de la pantalla (la cifra normalmente es 2). Si escribes POKE 23659, 0, el ordenador se detendrá tan pronto como el mensaje intente dejarse ver.

Esto no lo puedes escribir como comando directo, sino que debes incluirlo en un programa. Por ejemplo, inténtalo con ésta breve demostración:

```
10 POKE 23659,0
20 PRINT AT 5,5; RND
30 GOTO 20
```

Ejecuta (RUN) el programa. Presiona BREAK y comprueba como el ordenador se bloquea.

Si empleas este método, ten cuidado de que el programa no necesite visualizar mensajes tales como INPUT? o *scroll?*, o sucederá lo mismo. En todo caso utiliza INKEY$ si necesitas introducir datos.

Una manera de prevenir que un programa se autoejecute es utilizar MERGE, en lugar de LOAD. Es de nuevo el manual quien te da una pista: no puedes mezclar (MERGE) *«bytes».* Por lo que si almaceneas (SAVE) tus programas como CODE (es decir, como *«bytes»),* frustrarás al pirata. Debes almacenar todo: las variables del sistema, todas las del BASIC, el programa y toda la memoria reservada —la que hay por encima del *buffer* de la impresora.

Así, el CODE al que comienza SAVE es 23552, que es el comienzo de las variables del sistema. El número de *bytes* es N-23552, siendo N cualquier número grande mayor de STKEND (la dirección de comienzo del espacio reservado). El manual proporciona esta dirección como PEEK 23653 + 256*PEEK 23654. Si deseas almacenar (SAVE) toda la RAM destinada al usuario, entonces haz N igual a 65535. Este es el máximo número de *bytes* que puedes almacenar en la cinta. Por tanto, dispón la siguiente línea en tu programa.

```
1 SAVE "NOMBRE DEL PROGRAMA"
  CODE 23552,N-23552
```

Después añade una segunda línea con POKE, para hacer que el programa no funcione al utilizar BREAK:

Ahora teclea GOTO 1 y almacena el programa. El comando LOAD «NOMBRE DEL PROGRAMA» CODE causará la autoejecución del programa.

El comando LOAD ordinario no funcionará si N es muy grande, por ejemplo, 65.000 en la máquina de 48 K. Es derrochar cinta y memoria y como hay espacio apenas suficiente en la memoria del ordenador para tu programa, no cabrá nada más (tal como un programa de copia en cinta): por lo que tu programa está seguro.

```
2 POKE 23659,0
```

P y R.

¿Cómo puedo cargar una rutina en código máquina desde el BASIC de tal manera que ésta permanezca transparente al usuario?

Puedas utilizar un cargador como el de a continuación:

```
10 CLEAR N-1
20 LOAD "fichero en codigo
   maquina" CODE
```

Siendo N la dirección inicial de la rutina en código máquina.

Almacena este programa utilizando SAVE «nombre del cargador» LINE 10 para que se autoejecute. Después, tras él en la cinta, almacena la rutina en código máquina con SAVE «fichero en código máquina» CODE N, B, donde B es el número de *bytes* requerido.

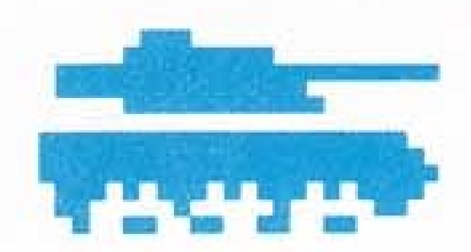


ACELERA TUS RUTINAS DE JUEGOS

El código máquina no es sólo para el programador experimentado. Tú puedes empezar con él, recurriendo a rutinas cortas que aceleren tus juegos en BASIC, y divertirte mientras aprendes.

La programación en código máquina dista mucho de ser fácil, al menos a simple vista. Para la mayoría de los usuarios de ordenadores personales, el código máquina no es más que un espantoso montón de números.

La mejor manera de comenzar es meterse a fondo con algunas rutinas escritas en código máquina. Así puedes familiarizarte, desde el principio, con sus ventajas y con lo que puede ofrecer a tus programas. Es por ello que hemos decidido llevarte a través de esos números misteriosos en este y en otros próximos artículos, de tal





manera que puedas ser capaz de desarrollar programas en código máquina por ti mismo.

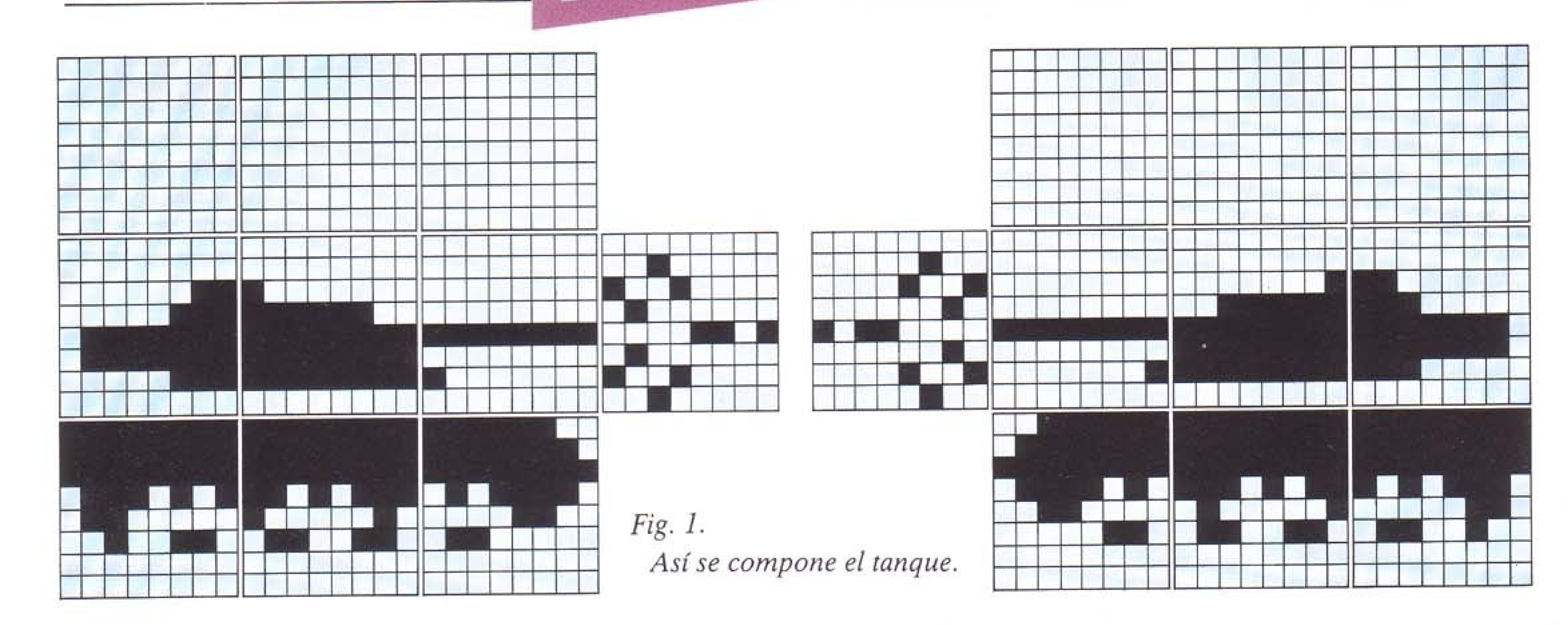
Los gráficos que van en este artículo utilizan programas en BASIC para depositar las rutinas en código máquina en la memoria de tu ordenador. De esta manera puedes producir movimientos mucho más rápidos y con más apariencia de vida de lo que podrías conseguir utilizando solamente el BASIC.

Para poner a punto los gráficos que aparecen en las figuras de los tanques y las ranas necesitarás hacer tres cosas:

Primera. Debes establecer una trama, o rejilla, en la memoria del ordenador, que define el tamaño del gráfico que deseas. Al principio, este entramado será reproducido como una serie de caracteres UDG (gráficos definidos por el usuario).

Segunda. Debes introducir un programa que te permita desplazar el entramado por toda la pantalla.

Tercera. Debes quitar los UDG existentes en el entramado, reemplazándolos por el gráfico que realmente deseas, sea un tanque, una rana o cualquier otra cosa.


Fig. 1.
Así se compone el tanque.

DEFINICION DE CARACTERES

Un carácter gráfico definido por el usuario, o carácter UDG para abreviar, es una letra (por ejemplo, la A) que puedes «moldear», consiguiendo otra cosa.

Cada letra consiste en 64 puntos (unos de un color, otro de otro) que forman una rejilla de 8 × 8. Suponiendo que puedes cambiar el interior de la rejilla, puedes escribir un programa que altere cada letra dándole la conformación que más te guste.

El **Spectrum** te proporciona 21 de tales letras: de la A a la U incluidas. Como puede verse en la figura 1, es factible disponer varias de estas pequeñas rejillas juntas, obteniéndose figuras de mayor tamaño. De una sola vez es posible tener, de esta forma, dos entramados de 3 × 3 (dejando tres caracteres UDG por encima) o conco de 2 × 2.

PUESTA A PUNTO DE LA REJILLA

Tanto el tanque como la rana ilustrados en este artículo necesitan un forma de 3 × 3 caracteres UDG, como vemos a continuación:

A B C
D E F
G H I

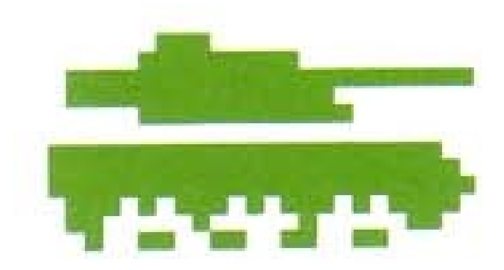
Una forma de realizarlo sería utilizar una serie de sentencias PRINT AT en BASIC:

```
PRINT AT 10,10;<graphics ABC>
PRINT AT AA,10;<graphics DEF>
```

Una manera mejor consiste en utilizar la rutina en código máquina que pone a punto el programa en BASIC presentado más abajo. Cuando lo teclees asegúrate de que estás utilizando las **palabras clave** del **Spectrum** y no escribas letra por letra esas palabras. Por ejemplo, CODE se consigue pulsando al mismo tiempo CAPS SHIFT y SIMBOL SHIFT, presionando la tecla I posteriormente.

```
10 IF PEEK 23733=127 THEN
   CLEAR 32399: LET
   B=32400: LET Z=0
20 IF PEEK 23733=255 THEN
   CLEAR 65199: LET
   B=65200: LET Z=1
30 FOR N=B TO B+129: READ A:
   POKE N,A: NEXT N
40 IF Z=1 THEN POKE
   65258,178: POKE
   65259,254: POKE
   65277,179: POKE
   65278,254
50 SAVE "Cuadrícula" CODE
   B,130
60 STOP
100 DATA 24,55,1,22,0,0,32,
    32,32,22,0,0,32,32,32,
    22,0,0,32,32
110 DATA 32,22,0,0,144,145,
    146,22,0,0,147,148,149,
    22,0,0,150,151,152,22
120 DATA 0,0,153,154,155,22,
    0,0,156,157,158,22,0,0,
    159,160,161,58,146,126
130 DATA 254,1,1,18,0,40,8,
    56,4,203,33,24,2,14,0,
    221,33,147,126,221
140 DATA 9,58,137,92,71,62,
    24,144,221,119,1,60,221,
    119,7,60,221,119,13,58
150 DATA 136,92,71,62,33,
    144,221,119,2,221,119,8,
    221,119,14,221,229,62,2,
    205
160 DATA 1,22,209,1,18,0,
    205,60,32,201
```



Puedes sentir que esta rutina es una manera enrevesada de escribir el equivalente a tres sentencias PRINT AT. Es cierto, pero obtenemos dos grandes ventajas:

1. Una vez que se ha tecleado completo, se puede guardar (SAVE) y utilizarlo cuantas veces gustes.

2. Durante su uso, esta rutina hará que todo el entramado (o el gráfico con que lo reemplacemos) sea desplazado por toda la pantalla mucho, mucho más deprisa de lo que es posible con su equivalente en BASIC.

Sin conocer el código máquina no puedes, naturalmente, entender cada uno de los números comprendidos en los DATA del programa. Pero resumiendo, las líneas 10 y 20 definen dónde se almacenará el programa. Si el tuyo es un **Spectrum** de 16 Kbytes, el programa se almacena en una parte diferente de la máquina que en el modelo de 48 K. Ten en cuenta que en algunos modelos de 48 Kbytes que primitivamente eran de 16 K, pero ampliadas, esto puede no funcionar. Si tienes un problema con un modelo de 48 K, omite la línea 10 y sustituye la línea 20 por:

```
20 CLEAR 65199: LET B=65200:
   LET Z=1
```

La línea 30 propone al ordenador que memorice los números que va sacando de las sentencias DATA.

La línea 50 es el comando SAVE. Puedes cambiar el nombre del fichero por otro que te guste más.

Las líneas 100 a 160 contienen los números que se transforman en tu rutina, en código máquina, una vez dispuestos correctamente en la memoria.

El SAVE de este programa se realiza de una manera diferente de la habitual. Cuando hayas tecleado el programa y lo ejecutes (RUN), el ordenador visualizará un mensaje que te invita a poner en marcha tu dispositivo de almacenamiento en cinta y pulsar después «cualquier tecla», para guardar la rutina de modo permanente.

DESPLAZAMIENTO DE LA REJILLA

Una vez que el programa esté depositado en la memoria (o cargado desde la cinta con LOAD y después ejecutado), está listo para ser utilizado. Simplemente recurre a NEW y [ENTER], para que el nuevo programa no corrompa al antiguo. Después teclea:

```
20 LET print=32400: LET
   B=32402: IF PEEK
   23733=255 THEN LET
   print=65200: LET B=65202
90 BORDER 0: PAPER 0: INK 4:
   CLS
100 LET Y=8: LET X=15: LET
    Y1=8: LET X1=15: LET Z=1
110 LET A$=INKEY$
120 IF A$="z" AND X>0 THEN
    LET X1=X-1: LET Z=1
130 IF A$="x" AND X<29 THEN
    LET X1=X+1: LET Z=2
140 IF A$="p" AND Y>0 THEN
    LET Y1=Y-1
150 IF A$="l" AND Y<18 THEN
    LET Y1=Y+1
170 LET X=X1: LET Y=Y1
180 PRINT AT Y,X;: POKE B,Z:
    RANDOMIZE USR print
190 GOTO 110
```

Advierte que las letras que van en minúscula deben ser tecleadas letra a letra, que no es lo mismo que escribir PRINT con solo la tecla P.

Este programa en BASIC te permite mover el entramado por toda la pantalla, recurriendo a las teclas: P, hacia arriba; Z, izquierda; L, hacia abajo y X, derecha. Cuando ejecutes este programa verás que no has creado un entramado de 3 × 3, sino dos, consistiendo el segundo en las letras JKLMNOPQR. Esto te permite disponer de dos versiones de tu gráfico final-uno que mira hacia la derecha cuando se desplaza en ese sentido y otro orientado a la derecha en el caso contrario.



Imagen ampliada.

También advertirás algo más: a medida que se desplaza el entramado, queda tras él una pista de caracteres no deseados. Esto se solventa con:

```
160 PRINT AT Y,X;: POKE B, 0:
    RANDOMIZE USR print
```

Así producimos un entramado de dimensión 3 × 3, formado por espacios, que borra todo lo que haya en la pantalla en la posición Y, X.

CREA UN TANQUE

Para convertir los entramados existentes en un tanque, basta con teclear estas pocas líneas con sentencias y DATAs. En ellas se define un tanque que apunta a la izquierda en el entramado 1 y a la derecha en el 2.

```
10 FOR N=USR "a" TO USR
   "r"+7: READ A: POKE N,A:
   NEXT N
1000 DATA 0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0,0
1010 DATA 0,0,0,0,0,0,0,0,
     0,0,0,0,255,0,1,0
1020 DATA 0,0,1,63,255,255,
     255,0,0,0,192,224,254,
     254,224,0
1030 DATA 63,127,255,122,48,
     6,0,0,255,255,255,235,
     65,102,0,0
```





WER

RE, S.A.

s. 232 24 61 - 232 25 52
ONA (SPAIN)



SOFTWARE

CENTURY SOFTWARE

¡¡¡ATENCION!!!

POWER, SOFTWARE, S.A. Perseguirá por todos los medios legales a su alcance, cualquier tipo de piratería, comercialización, importaciones o duplicados ilegales que sobre sus programas se practique.

FRENZY
SPECTRUM GAMES FROM
QUICKSILVA









BUG-BY
FRIDGE FRENZY
48K Spectrum

STRONTIUM DOG

Johny Alpha es un Agente Investigador Destructor, un cazador de recompensas del futuro. Johny lleva consigo Electro bengalas, que deslumbrarán a sus enemigos, obligándoles a dejar de disparar, y Bombas de Tiempo. Buena suerte en su Misión.

FRENZY

Gráficos en alta resolución a todo color, sonido, tabla de puntuaciones, pausa, cinco niveles de juego, robots, disparos Laser, monstruos asesinos, minas seguidoras explosivas, dificultad progresiva.



FRIDGE

Quienesquiera que hayan cargado la nave espacial de Freddie han provocado una gran confusión. Ahora Freddie está intentando volver a casa, pero el combustible radiactivo para el regreso se halla dispersado en su frigorífico.

QUINTIC WARRIOR

Parte hombre y parte superman, el guerrero Quintic se opone al siniestro hombre cangrejo y a las máquinas mutantes. ¿Estás dispuesto para el combate?

Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.

Nombre ____________

Dirección ____________

Teléfono ____________

Firma:

☐ Contrarreembolso. ☐ Adjunto Talón. ☐ Giro Postal.

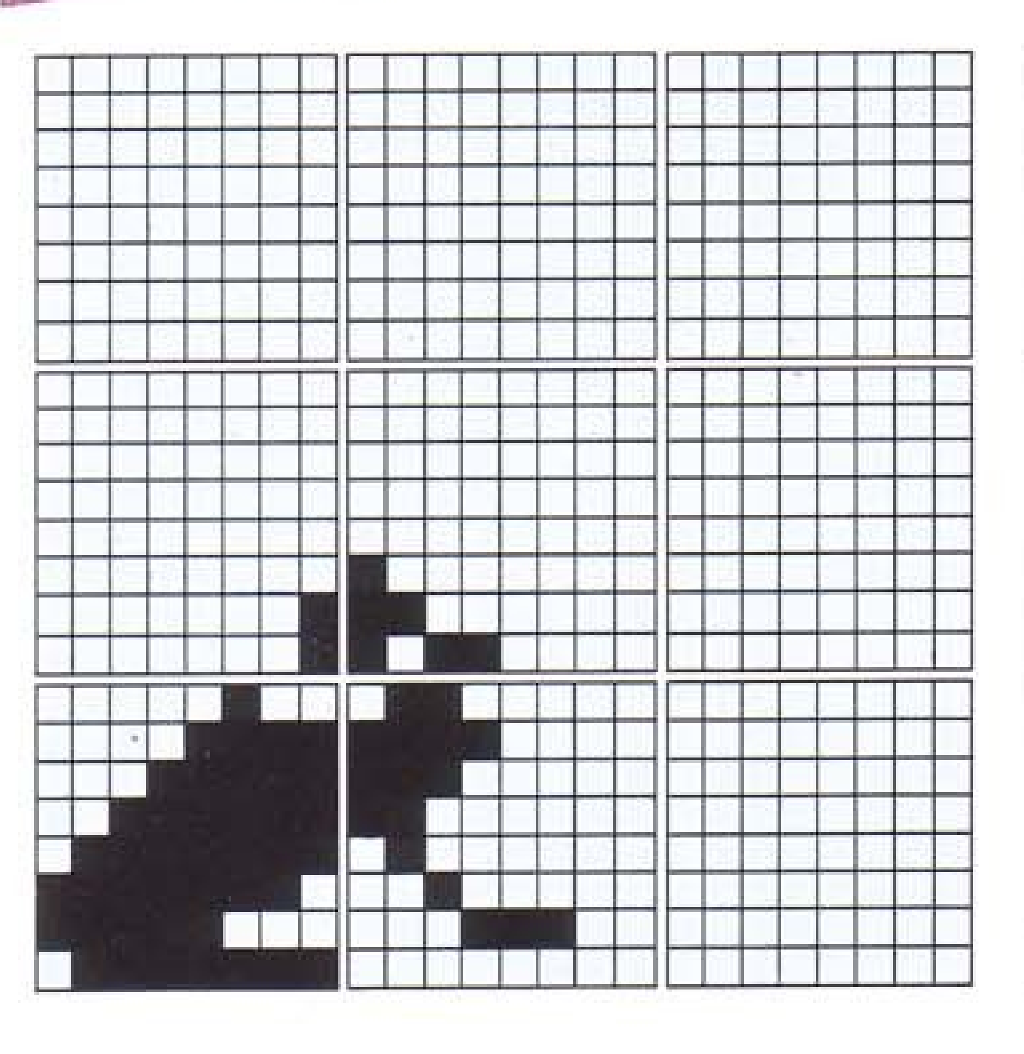
Deseo recibir información de sus programas en: MSX ☐ AMSTRAD ☐

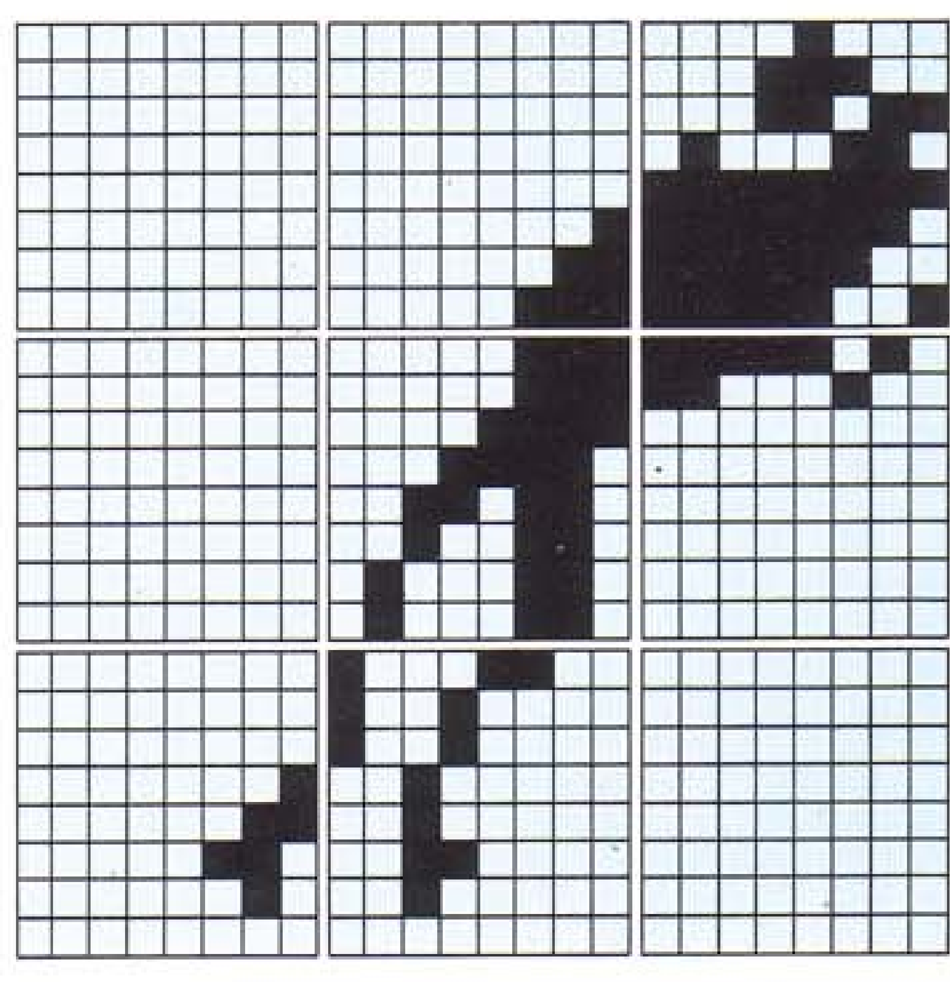
TITULO	SISTEMA	PRECIO UNIT.	CANTIDAD
LASER ZONE	Spectrum	975 Pts.	
GRIDRUNNER	Spectrum	975 Pts.	
FRENZY	Spectrum	1.275 Pts.	
ASTRO BLASTER	Spectrum	1.275 Pts.	
QUINTIC WARRIOR	Commodore	1.275 Pts.	
PURPLE TURTLES	Commodore	1.275 Pts.	
STRONTIUM DOG	Spectrum	1.800 Pts.	
STRONTIUM DOG	Commodore	2.100 Pts.	
FRIDGE FRENZY	Spectrum	1.800 Pts.	
BROAD STREET	Spectrum	2.400 Pts.	
BROAD STREET	Commodore	2.600 Pts.	

```
1040 DATA 255,255,255,174,6,
     100,0,0
1050 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0
1060 DATA 0,0,0,0,0,0,0,0,0,
     0,3,7,127,127,7,0
1070 DATA 0,0,128,252,255,
     255,255,0,0,0,0,0,255,
     0,128,0
1080 DATA 255,255,255,117,
     96,38,0,0,255,255,255,
     215,130,102,0,0
1090 DATA 252,254,255,94,12,
     96,0,0
```

Dos caracteres UDG más se encargan de «armar» al tanque. Presiona BREAK, y teclea después:

```
 10 FOR N=USR "a" TO USR
    "t"+7:READ A: POKE N,A:
    NEXT N
115 IF INKEY$="□" THEN GOTO
    200
200 IF Z=2 THEN GOTO 300
210 FOR N=X-1 TO 0 STEP-1
220 PRINT INK 5;AT Y+1,N;
    CHR$ 162
230 PAUSE 1
240 PRINT AT Y+1,N;"□"
250 NEXT N
260 GO TO 110
300 FOR N=X+3 TO 31
310 PRINT INK 5;AT Y+1,N;
    CHR$ 163
320 PAUSE 1
330 PRINT AT Y+1,N;"□"
340 NEXT N
350 GO TO 110
```





2. Diseño de la rana. Las rejillas se componen de 8 × 8 pixels.

```
1100 DATA 0,4,9,2,176,2,9,
     4,0,32,144,64,13,64,
     144,32
```

Ejecuta el programa y podrás disparar el tanque con la tecla SPACE.

CREA UNA RANA

Para convertir el entramado existente en una rana, primero tendrás que deshacerte, naturalmente, del tanque. Así que teclea NEW y después ENTER.

Lo que conseguimos con esto es borrar tanto el tanque como el programa que lo controlaba. El programa en código máquina que pone a punto los entramados seguirá, sin embargo, en la memoria del **Spectrum,** a menos que hayas desconectado el enchufe de alimentación.

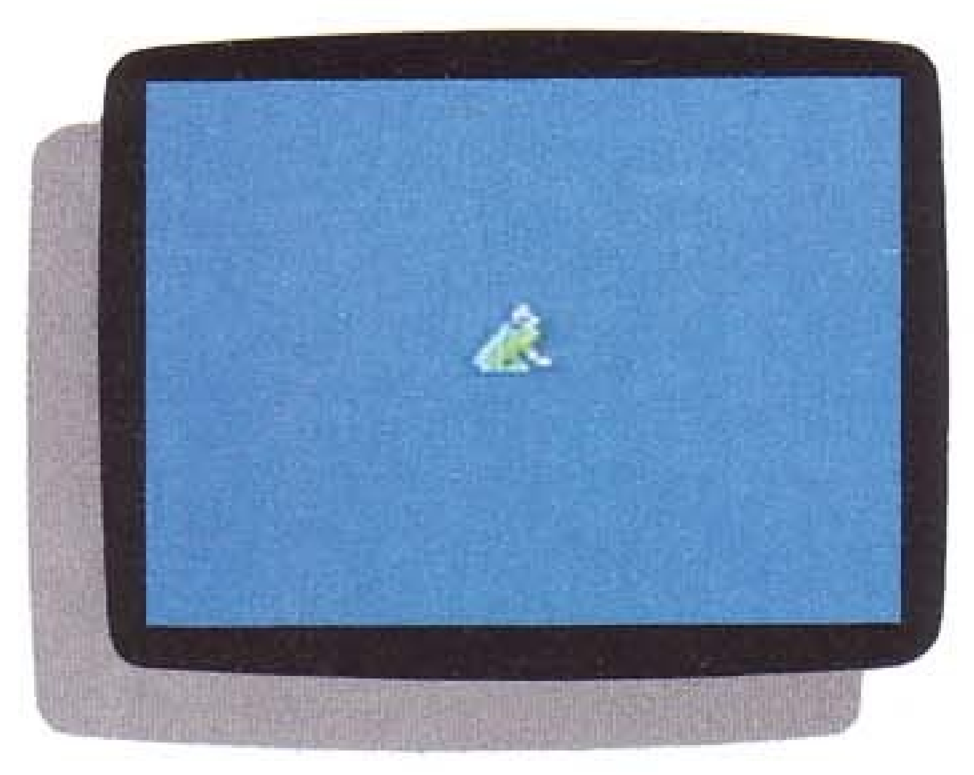
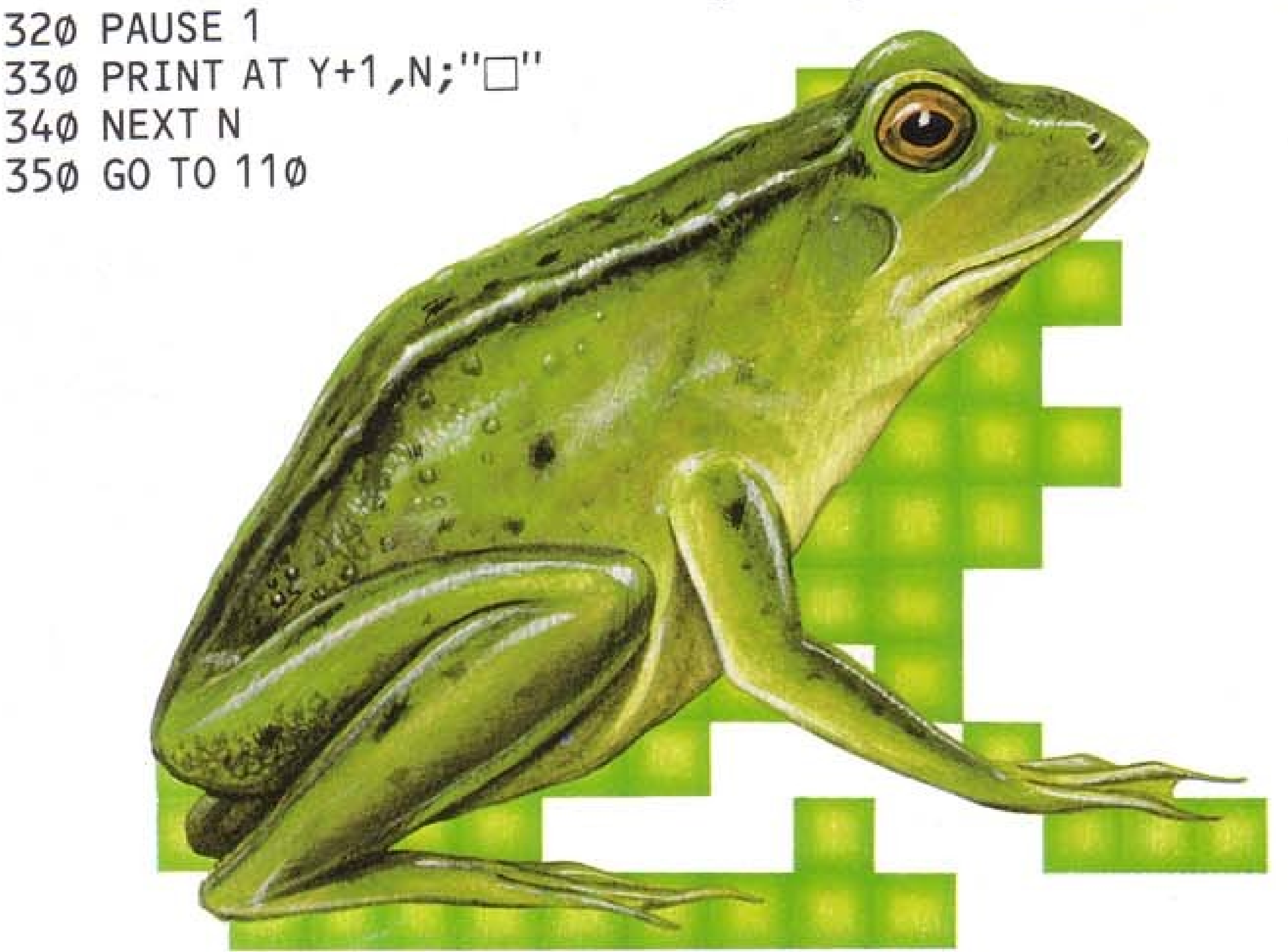


Imagen ampliada de la rana que se forma con este programa.



Si ya has desconectado la alimentación, cuando vuelvas a poner en marcha el ordenador teclea CLEAR 32399 en el modelo de 16 Kbytes o CLEAR 65199 en el de 48 K. Esto sirve para reservar un espacio destinado al código máquina en la parte más alta de la memoria de nuestro ordenador, allí donde el BASIC que escribiremos a continuación no podrá llegar y contaminarlo. Seguidamente, teclea LOADD""CODE y presiona la tecla *PLAY* de tu *cassette* para cargar la rutina en código máquina original.

Una vez almacenado el código má-

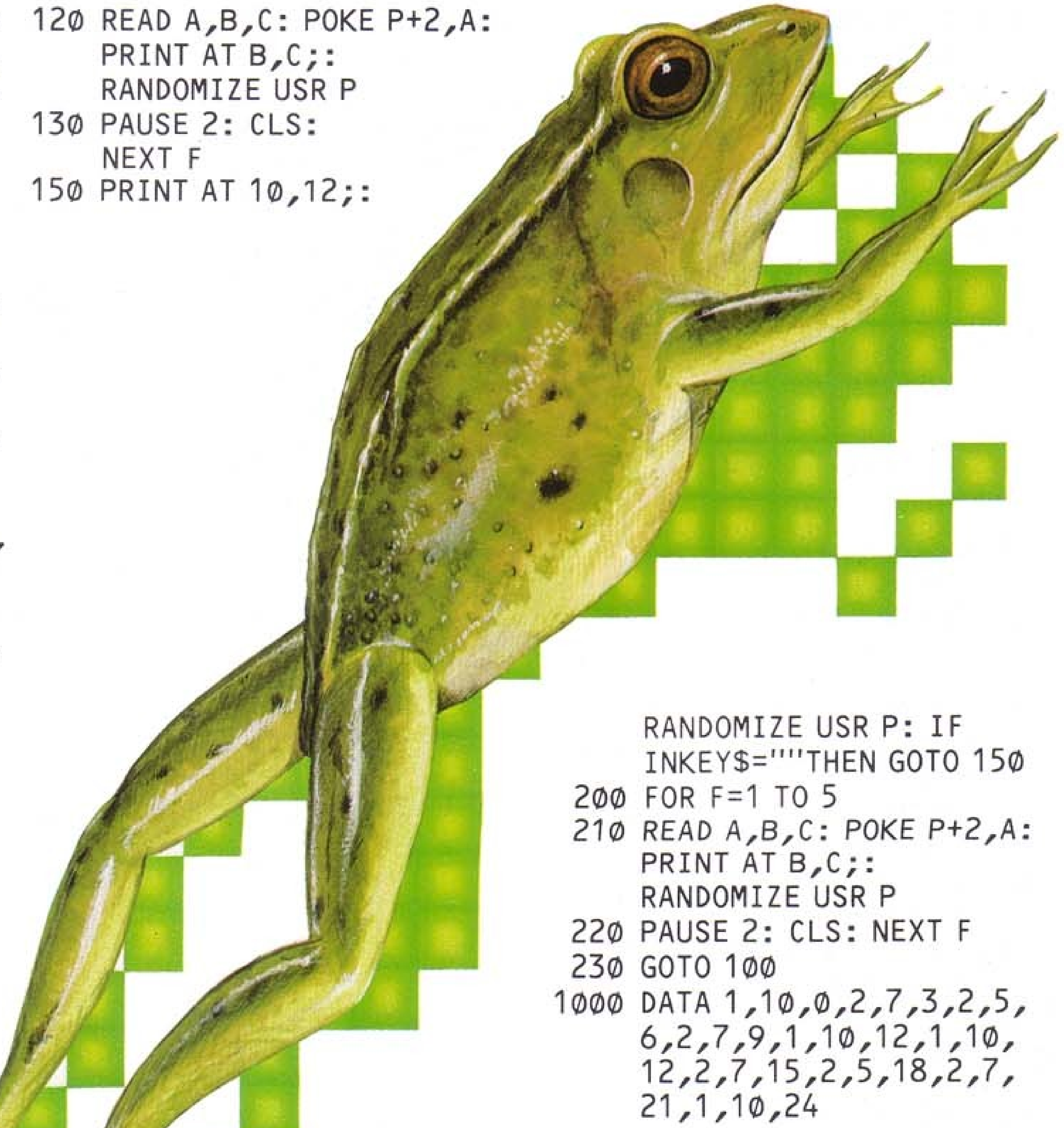
quina, teclea las siguientes sentencias más los DATA que definen a la rana. Entonces podrás ejecutar (RUN) el programa.

```
  30 RESTORE 5000: FOR N=USR
     "a" TO USR "r"+7: READ
     A: POKE N,A: NEXT N
5000 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0
5010 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,1,1
5020 DATA 0,0,0,0,0,128,192,
     176,0,0,0,0,0,0,0,0
5030 DATA 4,15,31,63,127,
     254,248,127,96,240,224,
     192,64,32,156,192
5040 DATA 0,0,0,0,0,0,0,0
5050 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,0,1,3,7
5060 DATA 8,28,27,70,255,
     254,252,249,0,0,0,0,0,
     0,0,0
5070 DATA 7,7,15,30,54,38,
     70,70,250,196,0,0,0,0,
     0,0
5080 DATA 0,0,1,1,3,6,2,0,
     140,144,16,32,32,48,
     32,0
5090 DATA 0,0,0,0,0,0,0,0
```

CONTROLANDO A LA RANA

Para controlar los movimientos de la rana, necesitas:

```
  10 BORDER 0: PAPER 0: INK
     4: BRIGHT 1: CLS
  20 LET P=32400: IF PEEK
     23733=255 THEN LET
     P=65200
 100 PRINT AT 10,0;:
     RANDOMIZE USR P: IF
     INKEY$=""THEN GOTO 100
 110 RESTORE 1000: FOR F=1
     TO 5
 120 READ A,B,C: POKE P+2,A:
     PRINT AT B,C;:
     RANDOMIZE USR P
 130 PAUSE 2: CLS:
     NEXT F
 150 PRINT AT 10,12;:
     RANDOMIZE USR P: IF
     INKEY$=""THEN GOTO 150
 200 FOR F=1 TO 5
 210 READ A,B,C: POKE P+2,A:
     PRINT AT B,C;:
     RANDOMIZE USR P
 220 PAUSE 2: CLS: NEXT F
 230 GOTO 100
1000 DATA 1,10,0,2,7,3,2,5,
     6,2,7,9,1,10,12,1,10,
     12,2,7,15,2,5,18,2,7,
     21,1,10,24
```





FRANKIE VA A HOLLYWOOD

Este es un juego de estrategia y habilidad. Nuestro protagonista está en un gran aprieto: ¡no tiene personalidad! Eres tú, con tu habilidad y la ayuda de **Frankie**, el que tienes que darle los puntos necesarios para ir proporcionándole personalidad mediante investigaciones en las pantallas por donde vas a pasar.
Durante la primera parte encuentras cuatro casas similares (¡ten cuidado, porque parecen iguales!). Cada una de ellas tiene cuatro puertas de entrada. Una vez traspasada una de ellas, te encuentras en un vestíbulo con otras dos o tres puertas: una es por la que has entrado; otra, es el salón, y la tercera (que no aparece siempre) es la cocina. Estas cuatro casas —por las que irás pasando— significan el amor, el sexo, la muerte y el odio. Están representadas por símbolos en la parte inferior derecha de la pantalla, y, sobre cada uno de ellos, hay una columna cuya altura indica los descubrimientos que vas haciendo en cada casa.
Tu misión es investigar aquellos objetos que encuentres a tu paso, porque dan puntos para incrementar la personalidad. La investigación se realiza situándose delante del objeto y pulsando «fuego» con el *joystick* en posición «hacia arriba». Verás que tu personaje levanta el brazo y señala con un dedo el objeto en cuestión. Aparecerá en la pantalla una ampliación del objeto que estudias (armarios de cocina, cajones del salón...) con cosas que puedes llevarte como ayuda, entre las que encontrarás desde armaduras, vídeos, pescado..., hasta pistolas y corazones. Dispones de un inventario al que puedes acceder pulsando «fuego» y moviendo el *joystick* hacia atrás, obteniendo una pantalla con los objetos que llevas contigo. Para coger un objeto, lo único que tienes que hacer es señalar con la mano aquello que deseas o bien la salida. El objeto que hayas elegido aparecerá reflejado en tu inventario, que sólo podrá contener ocho objetos nada más. Cada objeto tiene su utilidad (¡no hemos encontrado todas!). El vídeo te sirve para ponerlo en la televisión que encontrarás en algunos salones cuando las investigues, entonces saldrá una pequeña pantalla con imágenes del vídeo que te servirán de ayuda.
Procura no liarte mucho con las casas, ya que éstas se comunican entre sí por las puertas traseras de las cocinas y es preciso que sepas en cuál de ellas te encuentras. Por eso, es recomendable que, conforme vayas avanzando por casas y habitaciones, hagas un plano de cómo puedes acceder de una casa a otra.



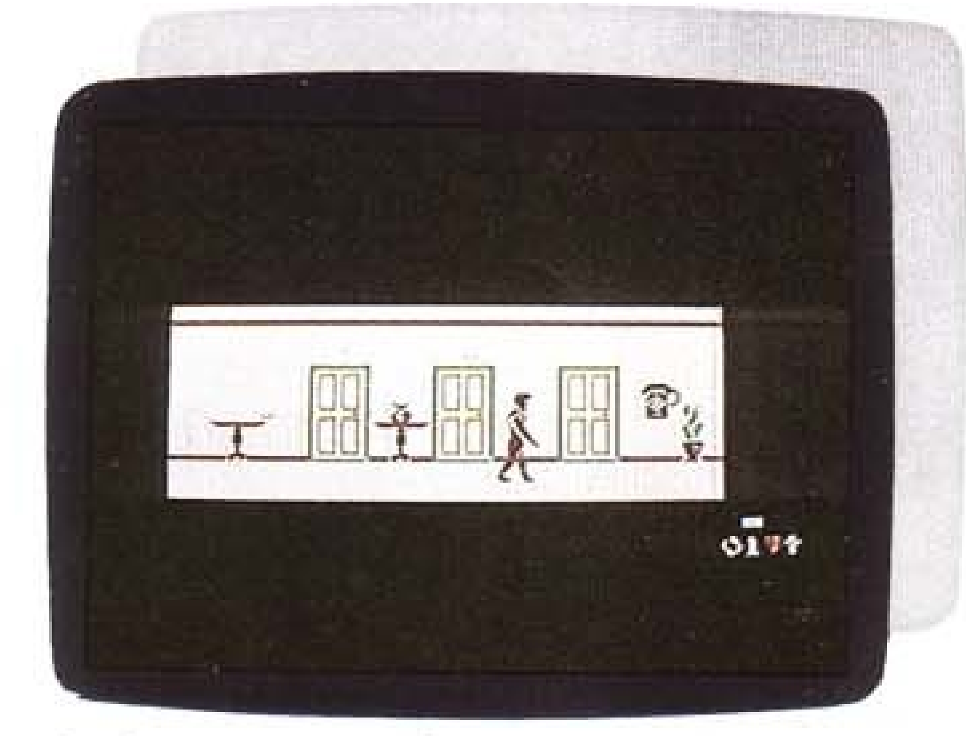
Tu misión es aumentar la personalidad.



El juego ofrece multitud de pantallas.



En cada habitación encontrarás objetos útiles.



Cada puerta te lleva a una nueva estancia.

DATOS GENERALES	CALIFICACION
TITULO: Frankie goes to Hollywood	GRAFICOS: 4 sobre 5
FABRICANTE: Ocean	COLOR: 4 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4,5 sobre 5
MEMORIA: 48 K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Aventura doméstica	REALISMO: 4 sobre 5

PARAR

TAB (c)

PRINT

P

CAMBIO

128K

SPECTRUM 128 EL SUMMUM

Spectrum, como líder, marca un nuevo hito en la historia de los ordenadores familiares.

El Spectrum 128.

Gran capacidad de memoria. Teclado y mensajes en castellano, teclado independiente para operaciones numéricas y de tratamiento de textos...

Sinclair e Investrónica han desarrollado una auténtica novedad. En ningún lugar del mundo, salvo en los Distribuidores Exclusivos de Investrónica, podrás encontrar el nuevo Spectrum 128.

Sé el primero en tener lo último.

SPECTRUM 128. NOVISIMUS



investronica

Tomás Bretón, 62.
Tel. (91) 467 82 10.
Telex 23399 IYCO E.
28045 Madrid

Camp, 80.
Tels. (93) 211 26 58 - 211 27 54.
08022 Barcelona

En una de las casas (¡adivina cuál!), al entrar en un salón, encontrarás un **muerto.** Ahora tu misión, además de aumentar la personalidad, es descubrir al terrible asesino mediante las pistas que te irán dando, conforme vayas pasando por las ditintas habitaciones. Estas pistas te van descubriendo datos sobre todo los posibles sospechosos del crimen, y también del asesino. Por eliminación, deberás quedarte sólo con aquel que puede cumplir las características del asesino. Cuando estés seguro, debes volver a la habitación donde encontraste al muerto, y aquí te aparecerá una lista con todos los personajes: tienes que señalar al asesino. Procura estar bien seguro, pues si te equivocas de individuo descenderá considerablemente el nivel de personalidad.
En uno de los muchos salones encontrarás una puerta, por la cual no puedes pasar, porque está cerrada. Necesitas una llave para abrirla. La llave la podrás encontrar en una de las habitaciones. ¡Búscala porque entrarás en una nueva etapa del juego!
Una vez dentro, existe un laberinto lleno de pasillos y con muchas salas diferentes. Pero esta vez no vamos a ser nosotros quien os descubra las pantallas restantes: esperamos que con estas indicaciones y vuestra habilidad consigáis llegar al final. ¡No os lo vamos a dar todo hecho...!

****

MARCIANOS EN LA AUTOPISTA

Una grave amenaza se cierne sobre la tierra: los extraterrestres nos han invadido y han instalado una base de operaciones desde la que pretenden extender su dominio.
Tú has sido elegido para detener el avance alienígena y conseguir la total destrucción de la «zona cero», lugar donde se encuentra establecida la base extraterrestre y alcanzar de este modo la salvación del planeta.
El objetivo final de esta misión, casi imposible, es detener el avance enemigo y destruir su centro neurálgico. Para llevar a cabo la misión cuentas con 5 *Vartons* programados para depositar el *Lasertron* en la base central enemiga, uno de los cuales, el *Vorton* principal, será el que dirija las operaciones y el único que puedes manejar.
Además del *Lasertron,* dispones también de un efectivo láser incorporado en el *Varton* principal que te permitirá destruir cuantos alienígenas salgan a tu encuentro, así como algunos de los obstáculos que se presenten.
Pero esto no es suficiente ya que precisarán también de un arma personal e intransferible sin cuyo concurso la misión no podrá realizarse, nos referimos a tu propio ingenio.
De la resolución de las distintas dificultades que surjan en la aventura dependerá el éxito o fracaso de la misión, y podemos asegurarte que necesitarás una estrategia preconcebida para alcanzar el objetivo marcado.
Para llegar a la «zona cero» tendrás que atravesar un total de 30 zonas con diversos grados de dificultad y distintos paisajes en sus márgenes.
Por otro lado, el camino a seguir por el *Lasertron* y los *Vartons* que lo protegen, está totalmente prefijado, es decir, tan sólo es posible mover por la pantalla el *Varton* principal que será el encargado de abrir el paso al *Lasertron.*
Puede ocurrir que mientras el *Varton* principal está en una zona, el resto de los *Vartons* y el *Lasertron* se encuentren en otra, lo cual no



Pantalla de presentación.



Empieza la dura misión.

DATOS GENERALES	CALIFICACION
TITULO: Highway Encounter	GRAFICOS: 4,5 sobre 5
FABRICANTE: Vortex Soft	COLOR: 4 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4,5 sobre 5
MEMORIA: 48 K	INTERES: 4 sobre 5
CLASE DE PROGRAMA: Marcianos en 3-D	REALISMO: 4 sobre 5

impide que éstos sean destruidos por los extraterrestres.
Las dificultades no acaban aquí ya que la misión cuenta con un tiempo limitado para ser realizada, que viene indicado en la parte inferior de la pantalla, que también informa sobre las puntuaciones más altas alcanzadas.
El juego está realizado en tres dimensiones y la construcción de las pantallas y el movimiento son francamente buenos, con una cuidada presentación y un sencillo manejo.
Tanto los *Vartons* como los distintos seres que van apareciendo y los decorados de la aventura resultan muy atractivos con una extraordinaria resolución gráfica.
En definitiva, se trata de un excelente juego creado por **Vortex Software** que seguro te hará pasar un buen rato delante de tu **Spectrum.**



El paisaje cambia y aumenta la dificultad.

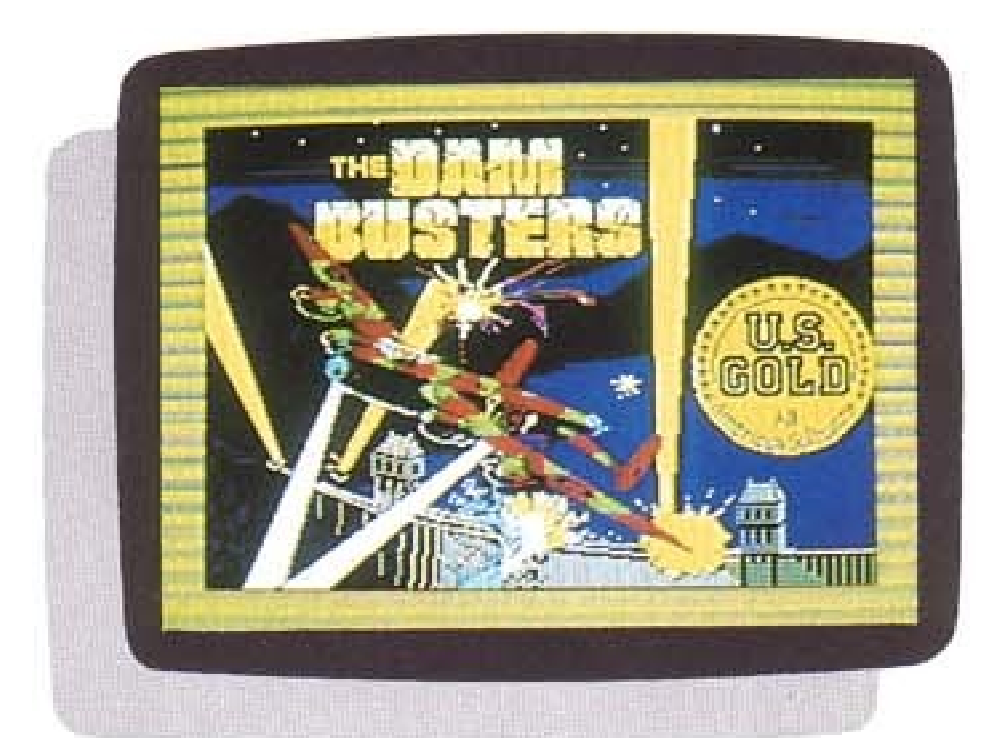


La zona 9, muy cerca del final.

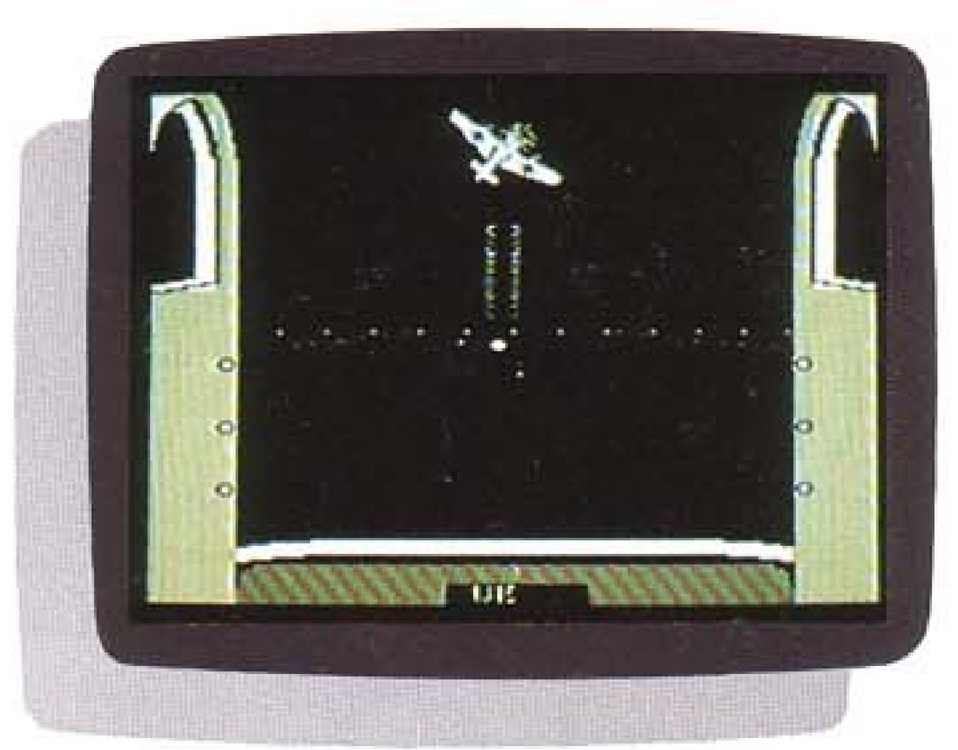
****

MISION SUICIDA

Durante la Segunda Guerra Mundial se plantearon determinados objetivos, tales como ataques a poblaciones, zonas industriales, etc., y también, cómo no, ataques a las presas. La segunda vía de agua más importante de Alemania era el río Oder tanto en cuanto al tráfico fluvial que en él se llevaba a cabo como en la faceta de aprovechamiento del agua para fines industriales (siderurgia pesada...).
Por tanto, el aniquilamiento de tales presas era un objetivo militar de suma importancia para los objetivos de los aliados.
Dam Buster recrea esa realidad histórica y nos devuelve a la situación en la que se vieron inmersos los tripulantes de los aviones **MARK III** durante los ataques que se llevaron a cabo en la presa mencionada.
Antes de comentar el juego hay que insistir en la promoción que se dio en su presentación en Gran Bretaña: La **U. S. GOLD** (distribuidora del mejor *software* procedente de Estados Unidos en Europa) lo presentó como el juego más conseguido que distribuía hasta la fecha (no olvidéis que entre otros, ya comercializa el **Beach Head I** y **II**, **Zaxxon...**), y esto explica la gran expectación existente entre el público.
El juego permite tres modalidades de juego: I) Práctica de aproximación a la presa. II) Teniente de vuelo. III) Jefe de escuadrilla.
La primera opción te permite entrenarte en tu misión de bombardeo de la presa. Para entenderlo hay que explicar que la misión de los **Dam** era un tanto extraordinaria: tenían que lanzar una bomba especialmente diseñada, a una distancia, velocidad y altura

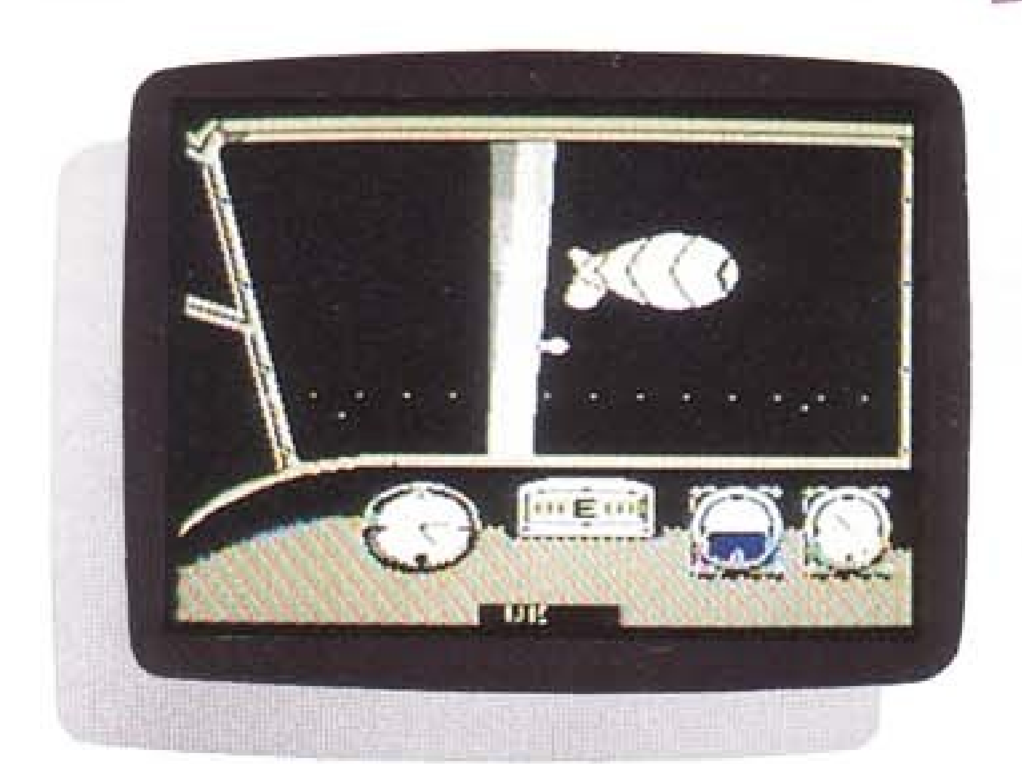


Pantalla de presentación.



Un ME-110 nos ataca por la cola.

DATOS GENERALES	CALIFICACION
TITULO: The Dam Busters	GRAFICOS: 4,5 sobre 5
FABRICANTE: U. S. Gold	COLOR: 3,5 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 5 sobre 5
MEMORIA: 48 K	INTERES: 5 sobre 5
CLASE DE PROGRAMA: Simulador bélico	REALISMO: 5 sobre 5



Globos y reflectores antiaéreos.

de la presa determinada para que ésta rebotase encima de una malla protectora antitorpedos, alcanzase el muro de la presa y se hundiera por fin en el agua al pie del muro, estallando en este lugar para causar el mayor daño posible. Esta opción del juego te coloca en una posición muy cercana a la presa y sólo tendrás que entrenarte para la misión de lanzamiento de la bomba.
La segunda opción del juego (Teniente de vuelo) hará que empieces tu misión encima del Canal de la Mancha.
Tal y como sucedía en la realidad el avance de tu avión se verá dificultado por aviones alemanes **ME.110,** reflectores antiaéreos y globos Cualquiera de estas tres amenazas tiene un antídoto de peso: una ametralladora del calibre 303; el juego consta de seis pantallas, que aparecen según indiques al ordenador dónde quieres colocarte en el avión: tienes cinco posiciones: piloto, artillero delantero, artillero trasero, navegante e ingeniero de vuelo.
Además dispones de un mapa de Europa, que te indica la posición de tu avión, así como los diferentes atacantes que encontrarás en tu camino. A veces, mientras estás en posición de piloto, un avión enemigo realiza un ataque en tu flanco derecho. El ordenador te indicará esta situación de dos maneras. Por otra parte, oirás un sonido de disparos y, por otra, verás una casilla en la parte inferior de la pantalla en la que aparecerán números: éstos te indican que has de pasar a la pantalla a la que corresponden, por no existir un peligro inminente en la zona que representan.
La tercera opción de vuelo es muy similar a las dos anteriores, ya que la diferencia estriba en que tendrás que realizar la misión completa incluyendo el despegue.
Estamos ante un juego cuidado, en cuanto a gráficos y a realismo, el ataque de los aviones recuerda vagamente al **Beach-Head,** aunque hay que reconocer que está mejor logrado. La mayor novedad del juego reside en la velocidad de cambio de las diferentes pantallas y, por ello, es fácil jugar durante horas y horas sin parar, el perfeccionamiento del juego alcanza incluso la posibilidad de realizar maniobras para «quitarse» de encima los aviones enemigos.

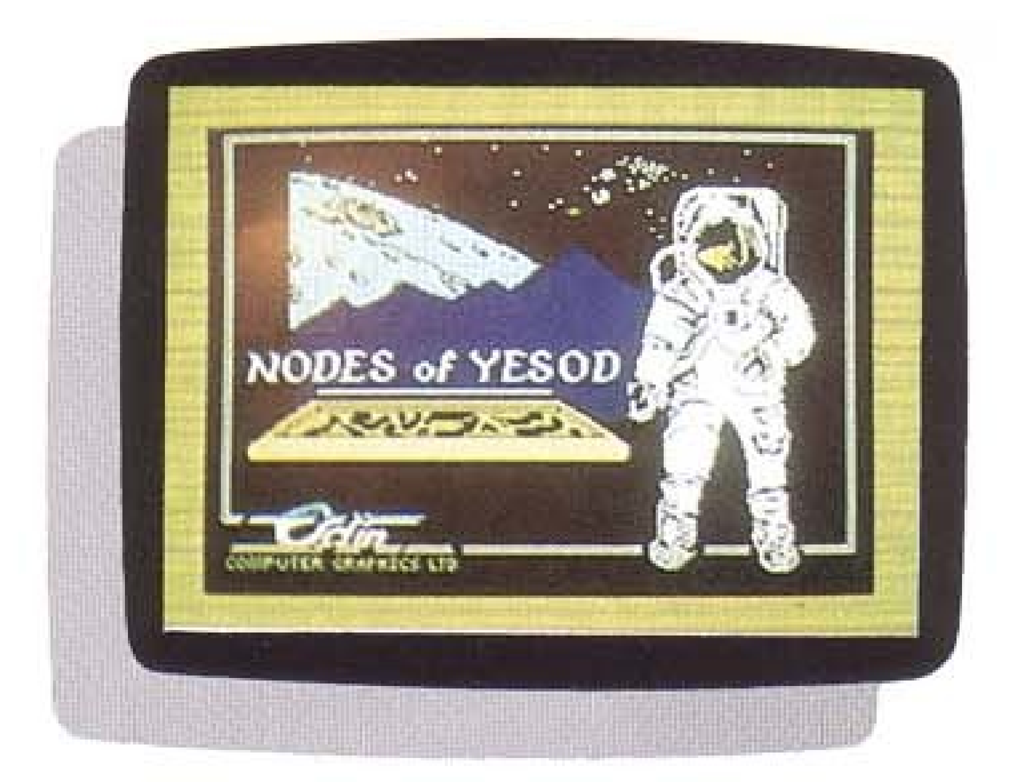
****

EXPLORACION LUNAR

A todos nos resultan familiares las imágenes de la superficie lunar. ¿Quién no ha visto en alguna ocasión sus escarpadas montañas y sus cráteres? Pero por el momento, nadie —al menos nadie que haya vivido lo suficiente para contarlo— conoce lo que encierran sus cavernas. Ahora tienes la posibilidad de explorarlas y desentrañar sus secretos junto con **Rt Hon Charlemagne Fortheringham-Grunes** («Charlie», para los amigos), uno de los aprendices de «Salvador del Universo». Es una ocasión única para alcanzar la gloria.
Nuestros científicos han descubierto que desde el centro de la Luna un emisor envía mensajes a algún lugar desconocido del espacio. Después de decodificarlo, se supo que informaban sobre la fragilidad de las defensas terrestres. Para evitar la destrucción de nuestro planeta tienes que encontrarlo y desactivarlo. No olvides que una vez en el interior de las cavernas no podrás recibir ninguna ayuda de fuera, por eso es conveniente que te prepares antes de entrar y te hagas con una especie de topo lunar.
Este animal tiene el poder de comer tierra y, por tanto, de abrir agujeros, en las paredes de las cavernas. Si consigues hacerte amigo de uno de ellos te facilitará la difícil misión; pero, por lo poco que se sabe de su forma de ser hay que decir que tienen un carácter voluble y no siempre están dispuestos a trabajar.
Del resto de la fauna y de la flora apenas se sabe nada. Esto, que por un lado constituye uno de los grandes alicientes de la expedición (puedes dar tu nombre a alguna de las especies lunares y pasar, esperemos que no a título póstumo, a la historia de la ciencia) es también el mayor, pero no el único, de los peligros. Por eso hay que ser prudente y evitar el trato con desconocidos.
También es posible que te encuentres con agentes enemigos dispuestos a

DATOS GENERALES	CALIFICACION
TITULO: Nodes of Yesod	GRAFICOS: 4 sobre 5
FABRICANTE: Odin Computer Graphics	COLOR: 4 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4,5 sobre 5
MEMORIA: 48 K	INTERES: 4 sobre 5
CLASE DE PROGRAMA: Fantasía espacial	REALISMO: 3,5 sobre 5

defender el emisor. Además en las cavernas pueden formarse una especie de tornados, que te expulsarán de donde estés. Y no olvides que, aunque la gravedad lunar sea menor que la de la Tierra, las caídas desde mucha altura son igual de mortales. Ten cuidado por donde andas. Por último, la falta de oxígeno es un problema: sólo dispones de tres cargas, una vez consumidas las tres se acabó todo. Estos son los peligros de los que podemos prevenirte. No sabemos si hay otros.



La pantalla de presentación.



Primera misión: cazar un topo lunar.



La Luna está bastante poblada.

Pero no todo van a ser dificultades. Algunas expediciones anteriores quizá hayan dejado cargas de oxígeno que, si consigues encontrarlas, aumentarán tu autonomía, y bastones gravitacionales con los que la marcha por las profundidades lunares sería mucho más fácil.
Para localizar el lugar donde se encuentra el emisor tienes que servirte de las ocho señales elementales de la alquimia que hallarás repartidas por las cavernas. ¡Suerte! La vida de la Tierra está en tus manos.

GYRON, EL HECHICERO CIENTIFICO

¿Has pensado alguna vez qué siente un ratoncillo cuando es introducido en un laberinto y debe encontrar su comida? Pues ahora es el momento de experimentar una sensación similar al iniciar esta apasionante aventura que se desarrolla en un laberinto tridimensional, y que tiene como objetivo localizar el Centro de la Sabiduría del hechicero científico. Tras muchos años de experimentos e investigaciones, este científico loco decidió conservar celosamente guardados todos sus descubrimientos antes de su muerte, para poder reiniciar su tarea en una vida futura. Para ello ideó este complicado y peligroso laberinto que se convierte en una verdadera fortaleza prácticamente inexpugnable.
Para alcanzar este recóndito lugar, el hechicero ha preparado diversas trampas con el objetivo de impedir el acceso a cualquier intruso a sus dominios.
La aventura consta de dos partes con distinto grado de dificultad: la primera de ellas, llamada Atrio, se desarrolla en un laberinto exterior, en tanto que la segunda, denominada Necrópolis, tiene como escenario un laberinto mucho más complejo y difícil, encontrándose en ella el Centro de la Sabiduría.
Para llevar a cabo la misión contamos con un robot de control remoto, armado con láser óptico de gran intensidad. Este robot no debe tocar los muros del laberinto, ya que cada impacto con ellos le ocasiona serios daños, que si sobrepasan el número 12 le destruirían totalmente, al igual que si nos estrellamos contra uno de estos muros con demasiada velocidad.
Además del rayo láser disponemos de un radar que nos indica nuestra posición en cada momento, pudiendo girar 180 grados en cualquier instante.
La pantalla muestra una vista frontal del laberinto. A nuestra derecha, tenemos un pequeño plano de la zona en que nos encontramos, unas líneas onduladas que nos indican los impactos que nos restan, el tiempo transcurrido, y por último, una figura geométrica que va cambiando de color a medida que atravesamos las 12 zonas de que consta el laberinto.
También aparecen en pantalla, a medida que iniciamos la marcha, unas bandas rojas y verdes en los laterales del monitor, que tienen distintos significados.
Las verdes indican el lugar en que nos encontramos y la dirección que tomará el rayo láser, mientras que las rojas, una vez sobrepasado el nivel, nos informan que podemos girar a izquierda o derecha sin chocar con el muro.
Por si fueran pocos los atractivos que ofrece este programa, creado por **Fire Bird,** también existe la posibilidad de ganar un sensacional **Porsche 924** o su equivalente en dinero, para aquellos que consigan descubrir el Centro de la Sabiduría creado por el hechicero científico.



Aquí se inicia la aventura dentro de la Necrópolis.



La pantalla muestra una sección del laberinto con una esfera acercándose.

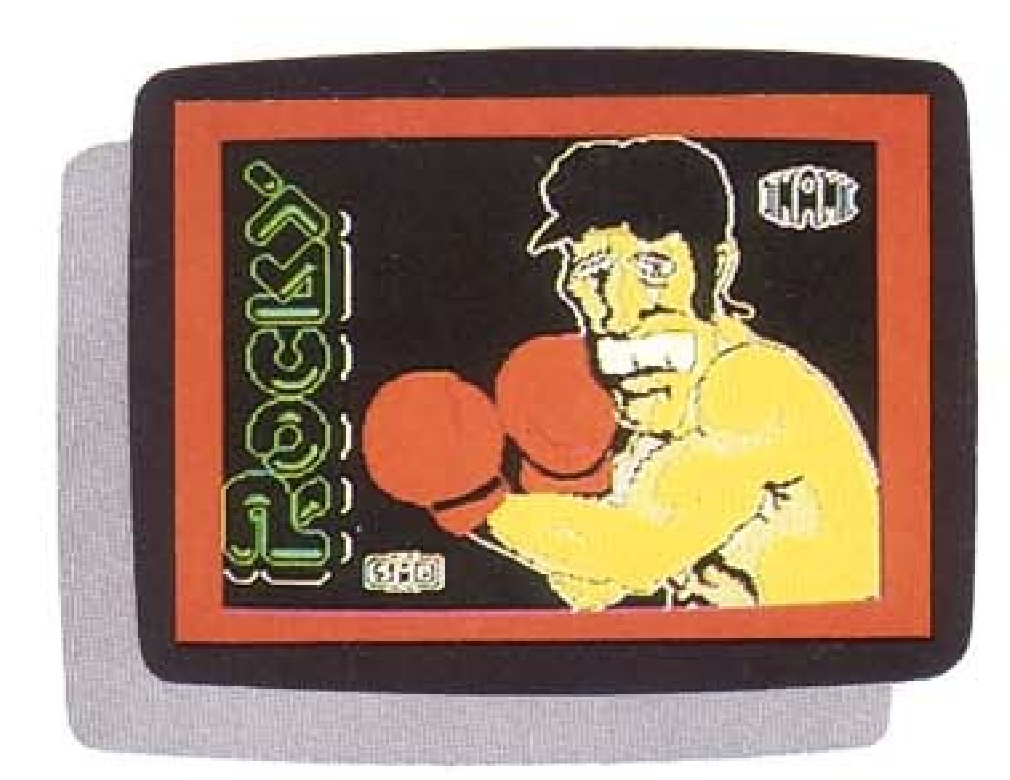


La gigantesca bola blanca nos impide continuar nuestro camino.

DATOS GENERALES	CALIFICACION
TITULO: Gyron	GRAFICOS: 3,5 sobre 5
FABRICANTE: Fire Bird	COLOR: 4 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4 sobre 5
MEMORIA: 48	INTERES: 3,5 sobre 5
CLASE DE PROGRAMA: Laberinto 3-D	REALISMO: 4 sobre 5

CONTRA LAS CUERDAS

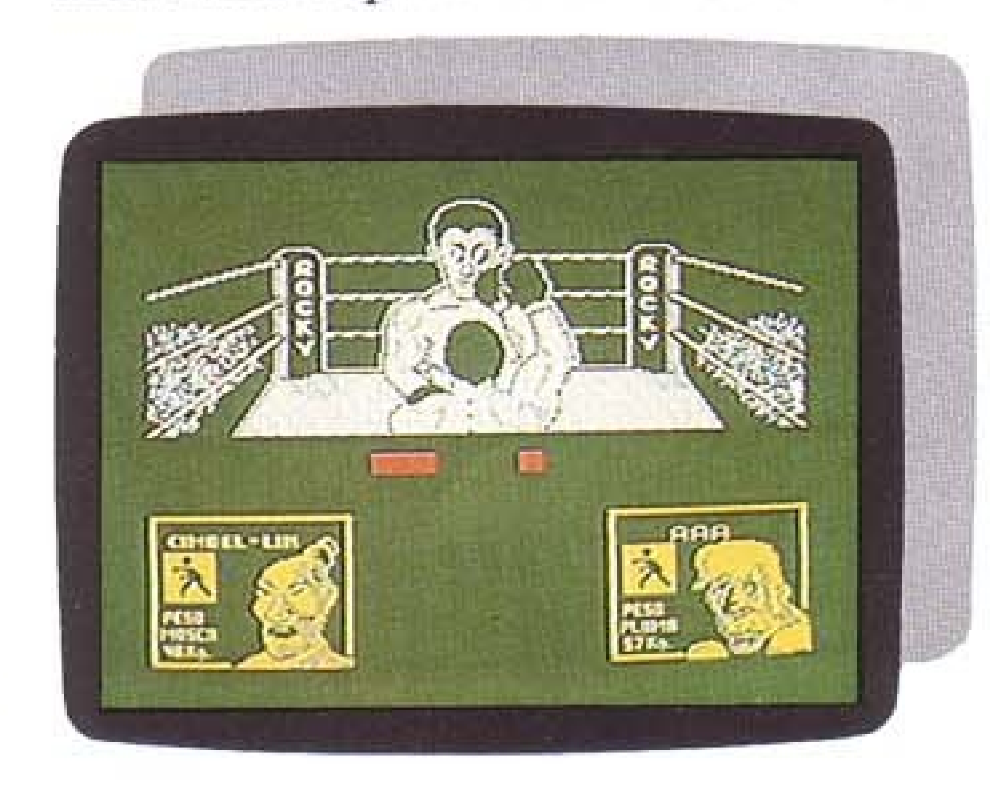
Miles de espectadores rodean el cuadrilátero, los púgiles se miran desafiantes, tratando de parecer tranquilos. Los nervios pueden traicionarles en cualquier momento. El público grita enfervorizado. El combate está a punto de comenzar. Si desea ser un temible boxeador capaz de derribar a cualquier contrincante con la fuerza de tus puños, este programa te permite hacerles frente sin que te duelan los golpes.
Tus ágiles reflejos deben estar atentos para esquivar los peligrosos ganchos que te envía tu rival y aprovechar cualquier ocasión para lanzar tus puños y alcanzar su férrea mandíbula.
La pantalla muestra un cuadrilátero visto desde un lateral, con los púgiles frente a frente y el público entusiasmado con el desarrollo del combate.
El vencedor del *round* será aquel con consiga un mayor número de golpes en el rostro de su contrincante hasta que consiga derribarlo.
Cuando tu puño golpee al rival, este recibirá el impacto girando su rostro del que se desprenderán unas gotas de sudor que denotal el esfuerzo realizado.
Para conocer el desarrollo del *match,* debajo del *ring* figuran unos indicadores de fuerza que indican los impactos recibidos y conseguidos, o lo que es lo mismo, los golpes que faltan para derribarle o para mantenerte en pie.
El programa no sólo dispone de la posibilidad de golpear, sino que también cuenta con la opción de defensa ante el ataque de tu oponente.
Alcanzarás la victoria en el combate derribando tres veces a tu rival, lo que te permitirá pasar a la siguiente categoría hasta que tras vencer a los cuatro aspirantes consigas ser el campeón del mundo.
Cada vez que consigas vencer a un oponente, el paso a la siguiente fase viene marcado por el característico «gong».
En la parte inferior de la pantalla aparece la ficha de los dos púgiles con la categoría a la que pertenecen, el peso en kilogramos, el nombre y una fotografía de ambos.
Lógicamente, a medida que te enfrentas con oponentes más pesados, crecen las dificultades y para vencer al peso pesado Figther Bull necesitarás andar muy ágil y realizar un buen juego de brazos para noquearle.
Rocky es un programa hecho por programadores españoles de la firma **Armid Soft,** habiendo sido realizados los gráficos por **Dinamic Software.**
En resumen, se trata de un original programa realizado en España, en el que deberás tener mucho cuidado con los golpes bajos y pensar que en este caso no podrá salvarle la campana.



Tras la pantalla de presentación, ¡segundos fuera! A partir de ese momento tu habilidad con los puños determinará quién levanta los brazos.



Aunque aquí no existe la diversidad de pantallas de otros juegos, el diseño y los movimientos de los púgiles no dejan de ser interesantes.





DATOS GENERALES	CALIFICACION
TITULO: Rocky	GRAFICOS: 4 sobre 5
FABRICANTE: Dinamic	COLOR: 3 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 3,5 sobre 5
MEMORIA: 48 K	INTERES: 3 sobre 5
CLASE DE PROGRAMA: Deportivo	REALISMO: 4 sobre 5



EL ZOCO

Desearía intercambiar programas de todo tipo, tengo más de 60, estoy interesado por el «Spy Hunter» y el «Skool Daze».
Mandar lista prometo contestación, tengo programas muy buenos.

Rafael Marín Alcaide.
Rubio i Ors, 33, 3.°, 2.ª
Sant Boi (Barcelona).

• • • • •

Desearía entrar en contacto con usuarios del Spectrum, para intercambio de ideas, programas y para que me resolvieran muchas dudas.

Alfonso Jiménez Sánchez.
Muza, 43, 2.° A.
Mérida (Badajoz)

• • • • •

Intercambio programas para el Plus 16K o 48 K.

Antonio Bravo.
Avda. Ferrol, 1, 7.°
28029 Madrid.

• • • • •

Intercambio programas para el ZX Spectrum.

Sabino Liébana Salvador.
Avda. José Antonio, 7, 1.° Dch.
34002 Palencia.

• • • • •

Vendo ZX Spectrum Plus 64 K, TV Philips TX, 6 canales, 30 revistas especializadas en ZX, libros de programación. Precio, a convenir.

Pere. Tel. (972) 57 09 37.

• • • • •

Vendo videojuegos Intellivision con 3 cartuchos por 6.000 ptas. Vendo y cambio juegos para Spectrum y vendo «California» por 8.000 ptas., nueva.

Aitor Guisasda.
Telletxe, 10, 3.° A.
Tel. (94) 469 29 31.
Algorta (Guecho).

• • • • •

Estoy interesado en intercambiar programas para ZX Spectrum.

Mario Saénz de Santamaría
Río Ebro, 27, 2.° C.
Miranda de Ebro (Burgos).

• • • • •

Cambio, vendo programas para el Spectrum. Mandar lista.

Emilio Jesús Hellín Asensio
Vista Alegre, 29, entlo.
30007 Murcia.

• • • • •

Busco contactar con chicos-as que posean en ZX81, para intercambiar información.

José F. Medina.
Queipo de Llano, 20, 1.°
Castro Urdiales (Cantabria).

• • • • •

Tengo los mejores juegos del Spectrum 48K, y estoy interesado en intercambiarlos con gente de Madrid. Interesados llamar al teléfono 4 50 39 05. Preguntar por Toni.

• • • • •

Euzkalclub, del Spectrum. Unico fin: Incrementar nuestras listas de programas.

Euzkalclub.
Maiatzren Botd, 2, 6.° D.
Lejowa (Vizcaya).

• • • • •

Intercambio programas e información para el Spectrum.

Rafael Antón Sánchez.
Apartado de Correos, 451.
Elche.

• • • • •

Me gustaría contactar con usuarios de ordenadores Sinclair (Spectrum 48K).

Estoy interesado en programar en Código Máquina, y tengo varios programas para trabajar en este campo.

Albert Casals.
Amalia 16, ático.
Mataró (Barcelona).

• • • • •

Intercambio y venta de juegos Spectrum 48K. Novedades

Ignacio (93) 353 90 71.

• • • • •

Intercambio programas, ideas, trucos, información.

Manolo Cazorla Muñox.
Ctra. Valls, N.° 18-4-2.
Margarita de Montbui.
Tel. (93) 404 32 03.
Preguntar por Enrique.
Barcelona.

• • • • •

Me gustaría contactar con usuarios del Spectrum 48K para intercambiar información y programas.

Sergio Brotons Briva.
Prolongación Alfonso V, 63.
Tel. (965) 59 09 39.
Cocentaina (Alicante).

• • • • •

Vendo lápiz óptico D(ktronics con Interface y cassette demostración, garantía 4 meses, por sólo 4.000 Ptas.

Javier Delicado Molina.
Apdo. 365.
Murcia 30080.

• • • • •

Vendo ZX Spectrum 16 K, ampliación 48 K, interface, Joystick Kempston, cassette Sanyo 50, juegos comerciales y copiones, 4 libros Spectrum y múltiples revistas. Precio a convenir.

Preguntar por Eduardo o Jesús.
Tel. (983) 23 71 83.

• • • • •

SINTAX ERROR?

En la linea 120 de la pagina 8 del coleccionable, donde dice ...LET xx=x, debe decir ...LET x=xx